International Journal of Engineering Innovations in Advanced Technology
ISSN: 2582-1431 (Online), Volume-4 Issue-4, December 2022

Design Of Power And Area Efficient ECC Processor Using Reversible Technique

Charishmmaa G

MTech Student, Department Of ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.
Email: gunducherry4567@gmail.com

Dr.S. Kishore Reddy

Associate professor, HOD, Department Of ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.Email: kishorereddy416@gmail.com

V Guravaiah

Assistant professor, Department Of ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.
Email: : guravaiahvemuri@gmail.com

***Abstract*— An exportable application-specific instruction-set elliptic curve cryptography processor based on redundant signed digit representation is proposed. The processor employs extensive pipelining. Furthermore, an efficient modular adder without comparison and a high through put modular divider, which results in a short data path form a ximiNISTreductionscheme.Thepropoprocessorerforms single point multiplication employing points in affine coordinates in 2.26 ms and runs at a maximum frequency of 160 MHz in Xilinx Virtex 5 (XC5VLX110T) field-programmable gate array.**

***Key Words*- field-programmable gate array, modular divider, digit representation.**

I. INTRODUCTION

Reversible logic has presented itself as a prominent technology which plays an imperative role in Quantum Computing. Quantum computing devices theoretically operate at ultrahigh speed and consume infinitesimally less power. Research done in this paper aims to utilize the idea of reversible logic to break the conventional speed-power trade-off, thereby getting a step closer to realise Quantum computing devices. To authenticate this research, various combinational and sequential circuits are implemented such as a 4-bit Ripple-carry Adder, (8-bit X 8-bit) Wallace Tree Multiplier, and the Control Unit of an 8-bit GCD processor using Reversible gates. The power and speed parameters for the circuits have been indicated, and compared with their conventional non-reversible counterparts. The comparative statistical study proves that circuits employing Reversible logic thus are faster and power efficient. The designs presented in this paper were simulated using Xilinx 9.2 software.

Reversible logic is widely used in low power VLSI. Reversible circuits are capable of back-computation and reduction in dissipated power, as there is no loss of information [1]. Basic reversible gates are employed to achieve the required functionality of a reversible circuit. The uniqueness of reversible logic is that, there is no loss of information since there is one-to-one correspondence between inputs and outputs. This enables the system to run backwards and while doing so, any intermediate design stage can be thoroughly examined.

A. Reversible Logic Gates

Boolean logic is said to be reversible if the set of inputs mapped have an equal number of outputs mapped i.e. they have one-to-one correspondence. This is realized employing reversible gates in the designs. Any circuit having only reversible gates is

Manuscript received Oct 10, 2022; Revised Oct 25, 2022; Accepted Nov 4, 2022

capable of dissipating no power [2]. Goals of Reversible Logic: A. Quantum Cost: Quantum cost of a circuit is the measure of implementation cost of quantum circuits. More precisely, quantum cost is defined as the number of elementary quantum operations needed to realize a gate.

There are many reversible gates such as Feynman, Toffoli, TSG, Fredkin, Peres, etc [3]. As the universal gates in boolean logic are Nand and Nor, for reversible logic, the universal gates are Feynman and Toffoli gates.

B. Feynman Gate

Feynman gate is a universal gate which is used for signal copying purposes or to obtain the complement of the input signal.

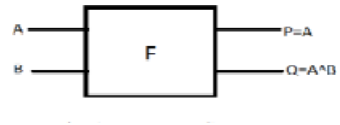

Figure 1 Feynman Gate

C. Fredkin Gate

It is a basic reversible 3- bit gate used for swapping last two bits depending on the control bit. The control bit here is A, depending on the value of A, bits B and C are selected at outputs Q and R. When A=0, (Q=B, R=C) whereas when A=1, (Q=C, R=B). Its block diagram is as shown in fig. 2:

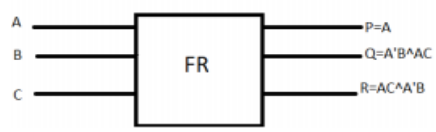

Figure 2 Fredkin Gate

D. Peres Gate

It is a basic reversible gate which has 3-inputs and 3-outputs having inputs (A, B, C) and the mapped outputs (P=A, Q=A^B, R=(A.B)^C). The block diagram is as shown in fig. 3

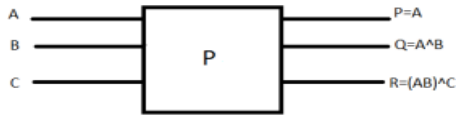

Figure 3 Peres Gate

E. Toffoli Gate

Toffoli gate is a universal reversible gate which has three inputs (A, B, C) mapped to three outputs (P=A, Q=B, R= (A.B)^C). The block diagram of Toffoli gate is shown in fig. 4

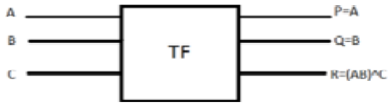

Figure 4 Toffoli Gate

F. TSG Gate

TSG gate is a reversible gate which has four inputs (A, B, C, D) mapped to four outputs (P=A, Q=A^B, R=A^B^D, S=(A^B)^D^AB^C). The block diagram of TSG Gate is shown in fig. 5

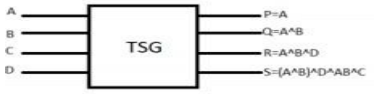

Figure 5 Tsg Gate

G. Design Of Control Unit For Gcd Processor

To illustrate the classical and reversible approaches to the Sequential Control Unit Design, reversible logic is employed for a special purpose processor that computes the GCD of two numbers. This GCD processor incorporates standard Euclid's Algorithm involving Subtract-Compare-Swap operation of two numbers. The basic principle is to subtract smaller of the two numbers repeatedly from the other number until we get the number that divides another [6]. A. Control Unit Control unit of GCD

processor generates the control signals to manipulate the operations in Data-path

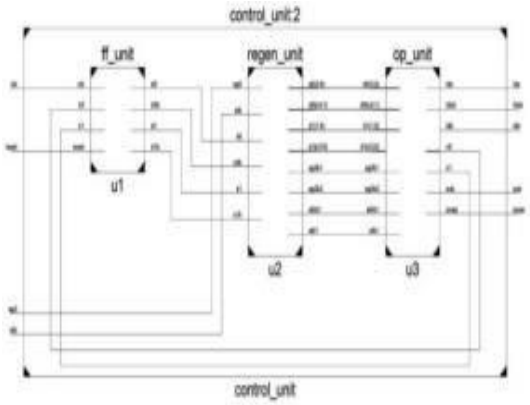

Figure 6 Block Diagram Of GCD Control Unit

H. Block Diagram Description

- *Flip-flop Module*

The control unit for GCD processor requires two Flipflops as binary state encoding is used for FSM. In this design reversible edge-triggered D Flip-flop is employed for state transitions [7]. Two D-latches are connected in Master-Slave mode to act as an edge-triggered D Flip-flop. Reversible D-latch is designed using Feynman and Fredkin gates [8]. RTL schematic of reversible D flip-flop obtained is shown in fig.

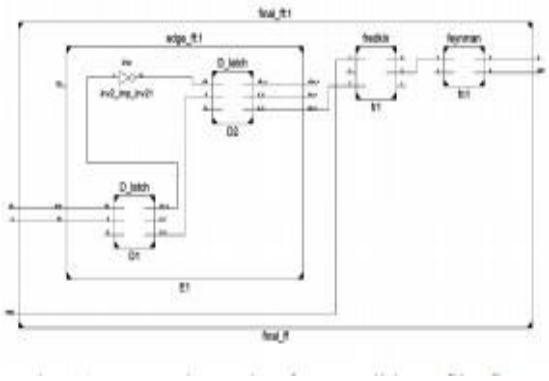

Figure 7 RTL schematic of Reversible D flip-flop

- *Regeneration Module*

To avoid multiple fan-out condition in the design, it is necessary to duplicate signals used for computation of output and next state. The duplication of input signals is achieved using Feynman gates.

- *Output Module*

The computation of the outputs and Next-state signals is done using reversible Fredkin gates. The functioning of output signals is driven by the algorithm. C. Final RTL schematic: The complete RTL schematic of GCD control unit is shown in fig.

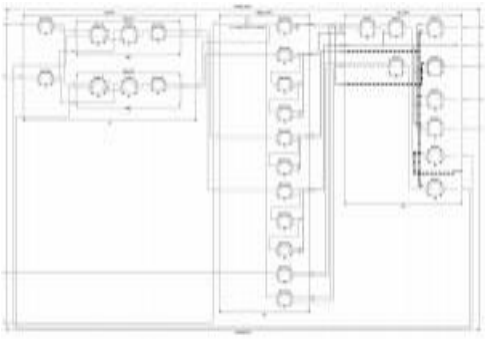

Figure 8 RTL Schematic Diagram Of Gcd Control Unit

I. Speed and power analysis:

Parameter (spartan3 xc3s50 family)	Irreversible GCD control unit	Reversible GCD control unit	Percentage improvement for reversible circuit
Speed(Max Clock freq)	434.33MHz	456.09MHz	5.01%
Power	25.02mW	24.19mW	3.31%

Table 1 Comparison of Reversible and Irreversible Control

J. Applications

Reversible logic design finds applications in various fields including Quantum computing, Nano-computing, optical computing, Quantum Computing Automata (QCA: study of mathematical objects called Abstract machines and the computational problems that can be solved using them), ultra- low power VLSI designing, Quantum dot cellular etc. The future of

computer chips is limited by Moore's law; hence an alternative is to build quantum chips. Our future research topic is designing a new reversible gate and to implement reversible logic into a complete Quantum processor capable of ultra-high speed and infinitesimally low power computing.

II. VLSI

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors.

A. ASIC

An Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use. For example, a chip designed solely to run a cell phone is an ASIC. Intermediate between ASICs and industry standard integrated circuits, like the 7400 or the 4000 series, are application specific standard products (ASSPs).

As feature sizes have shrunk and design tools improved over the years, the maximum complexity (and hence functionality) possible in an ASIC has grown from 5,000 gates to over 100 million. Modern ASICs often include entire 32-bit processors, memory blocks including ROM, RAM, EEPROM, Flash and other large building blocks. Such an ASIC is often termed a SoC (system-on-a-chip). Designers of digital ASICs use a hardware description language (HDL), such as Verilog or VHDL, to describe the functionality of ASICs.

Field-programmable gate arrays (FPGA) are the modern-day technology for building a breadboard or prototype from standard parts; programmable logic blocks and programmable interconnects allow the same FPGA to be used in many different applications. For smaller designs and/or lower production volumes, FPGAs may be more cost effective than an ASIC design even in production.

B. Language– Verilog

In the semiconductor and electronic design industry, Verilog is a hardware description language(HDL) used to model electronic systems. Verilog *HDL*, not to be confused with VHDL (a competing language), is most commonly used in the design, verification, and implementation ofdigital logic chips at the register-transfer level of abstraction. It is also used in the verification ofanalog and mixed-signal circuits.

Hardware description languages such as Verilog differ from software programming languages because they include ways of describing the propagation of time and signal dependencies (sensitivity). There are two assignment operators, a blocking assignment (=), and a non-blocking (<=) assignment. The non-blocking assignment allows designers to describe a state-machine update without needing to declare and use temporary storage variables (in any general programming language we need to define some temporary storage spaces for the operands to be operated on subsequently; those are temporary storage variables). Since these concepts are part of Verilog's language semantics, designers could quickly write descriptions of large circuits in a relatively compact and concise form. At the time of Verilog's introduction (1984), Verilog represented a tremendous productivity improvement for circuit designers who were already using graphical schematic capturesoftware and specially-written software programs to document and simulate electronic circuits.

A Verilog design consists of a hierarchy of modules. Modules encapsulate *design hierarchy*, and communicate with other modules through a set of declared input, output, and bidirectional ports. Internally, a module can contain any combination of the following: net/variable declarations (wire, reg, integer, etc.), concurrent and sequential statement blocks, and instances of other modules (sub-hierarchies). Sequential statements are placed inside a begin/end block and executed in sequential order within the block. But the blocks themselves are executed concurrently, qualifying Verilog as a dataflow language.

A subset of statements in the Verilog language is synthesizable. Verilog modules that conform to a synthesizable coding style, known as RTL (register-transfer level), can be physically realized by synthesis software. Synthesis software algorithmically transforms the (abstract) Verilog source into a net list, a logically equivalent description consisting only of

elementary logic primitives (AND, OR, NOT, flip-flops, etc.) that are available in a specific FPGA or VLSI technology. Further manipulations to the net list ultimately lead to a circuit fabrication blueprint (such as a photo mask set for an ASIC or a bit stream file for an FPGA).

C. System Verilog

System Verilog is a superset of Verilog-2005, with many new features and capabilities to aid design verification and design modeling. As of 2009, the SystemVerilog and Verilog language standards were merged into System Verilog 2009 (IEEE Standard 1800-2009).

The advent of hardware verification languages such as Open Vera, and Verisity's e language encouraged the development of Super log by Co-Design Automation Inc. Co-Design Automation Inc was later purchased by Synopsys. The foundations of Super log and Vera were donated to Accellera, which later became the IEEE standard P1800-2005: System Verilog.

The most valuable benefit of SystemVerilog is that it allows the user to construct reliable, repeatable verification environments, in a consistent syntax, that can be used across multiple projects

Some of the typical features of an HVL that distinguish it from a Hardware Description Language such as Verilog or VHDL are

- Constrained-random stimulus generation
- Functional coverage
- Higher-level structures, especially Object Oriented Programming
- Multi-threading and interprocess communication
- Support for HDL types such as Verilog's 4-state values
- Tight integration with event-simulator for control of the design

There are many other useful features, but these allow you to create test benches at a higher level of abstraction than you are able to achieve with an HDL or a programming language such as C.

III. PROPOSED METHOD

Redundant signed digits Elliptic curve cryptography (ECC) is an asymmetric cryptographic system that provides an

equivalent security to the well-known Rivest, Shamir and Adleman system with much smaller key sizes. The basic operation in ECC is scalar point multiplication, where a point on the curve is multiplied by a scalar. A scalar point multiplication is performed by calculating series of point additions and point doublings. Using their geometrical properties, points are added or doubled through series of additions, subtractions, multiplications, and divisions of their respective coordinates. Point coordinates are the elements of finite fields closed under a prime or an irreducible polynomial. Various ECC processors have been proposed in the literature that either target binary fields, prime fields, or dual field operations. In prime field ECC processors, carry free arithmetic is necessary to avoid lengthy data paths caused by carry propagation. Redundant schemes, such as carry save arithmetic (CSA), redundant signed digits (RSDs), or residue number systems (RNSs), have been utilized in various designs. Carry logic or embedded digital signal processing (DSP) blocks within field programmable gate arrays (FPGAs) are also utilized in some designs to address the carry propagation problem. It is necessary to build an efficient addition data path since it is a fundamental operation employed in other modular arithmetic operations. Modular multiplication is an essential operation in ECC. Two main approaches may be employed. The first is known as interleaved modular multiplication using Montgomery's method. Montgomery multiplication is widely used in implementations where arbitrary curves are desired. Another approach is known as multiply-then-reduce and is used in elliptic curves built over finite fields of Messene primes. Messene primes are the special type of primes which allow for efficient modular reduction through series of additions and subtractions. In order to optimize the multiplication process, some ECC processors use the divide and conquer approach of Karatsuba–Ofman multiplications, where others use embedded multipliers and DSP blocks within FPGA fabrics.

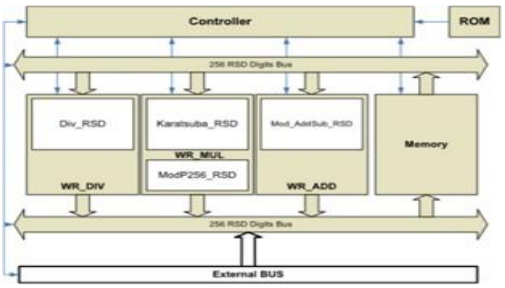

Figure 9 Overall Processor Architecture

External data enter the processor through the external bus to the 256 RSD digits input bus. Data are sent in binary format and a binary to RSD converter stuffs zeros in between the binary bits in order to create the RSD representation. Hence, 256-bits binary represented integers are converted to 512- bits RSD represented integers. To convert RSD digits to binary format, one needs to subtract the negative component from the positive component of the RSD digit.

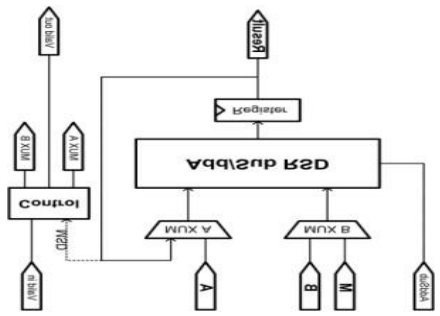

Figure 10 Modular Addition Subtraction Block Diagram

In order to overcome the problem of overflow introduced in the adder proposed in, a new adder is proposed based on the work proposed in. The proposed adder consists of two layers, where layer 1 generates the carry and the interim sum, and layer 2 generates the sum, as shown in Fig. 3. Table I shows the addition rules that are performed by layer 1 of the RSD adder, where RSD digits 0, +1, and −1 arerepresented by Z, P, andN, respectively. It works by assuring that layer 2 does not generate overflow through the use of previous digits in layer 1. The proposed adder is used as the main block in the modular addition component to take advantage of the reduced overflow feature. However, overflow is not an issue in both the multiplier and the divider when an RSD adder is used as an internal block. Hence, the reduced area is taken as an advantage in instantiating adders within the multiplier and the divider. The n-digits modular addition is performed by three levels of RSD addition. Level 1 performs the basic addition of the operands which produces n+1 digits as a result. If the most significant digit (MSD) of level 1 output has a value of 1/−1, then level 2 adds/subtracts the modulo P256 from the level 1 output correspondingly. The result of level 2 RSD addition has n+2 digits; however, only the n+1th digit may have a value of 1/−1. This assertion is backed up by the fact that the operation of level 2 is a reversed operation with the modulo P256, and most importantly, the proposed adder assures that no unnecessary overflow is produced. If the n+1th digit of level 2 result has a value 1 or −1, then level 3 is used to reduce the output to the n-digit range. Algorithm 3 shows the sequence of operations performed by the modular addition block. Notice that one modular addition is performed within one, two, or three clock cycles.

IV. CONCLUSION

a NIST 256 prime field ECC processor implementation in FPGA has been presented. An RSD as a carry free representation is utilized which resulted in short data paths and increased maximumfrequency. We introduced enhanced pipelining techniques within Karatsuba multiplier to achieve high throughput performance by a fully LUT-based FPGA implementation. An efficient binary GCD modular divider with three adders and shifting operations is introduced as well. Furthermore, an efficient modular addition/subtraction is introduced based on checking the LSD of the operands only. A control unit with add-on like architecture is proposed as a reconfigurability feature to support different point multiplication algorithms and coordinate systems. The implementation results of the proposed processor showed the shortest data path with a maximum frequency of 160 MHz, which is the fastest reported in the literature for ECC processors with fully LUT-based design. A single point multiplication is achieved by the processor within 2.26 ms, which is comparable with ECC processors that are based on embedded multipliers and DSP blocks within the FPGA. The main advantages of our processor include the exportability to other FPGA and ASIC technologies and expandability to support different coordinate systems and point multiplication algorithms.

REFERENCES

[1] Landauer, Rolf, "Irreversibility and heat generation in the computing process," IBM Journal of Research and Development ,vol.44, no.1.2, pp.261,269, Jan. 2000

doi: 10.1147/rd.441.0261

[2] Bennett, C.H., "Logical Reversibility of Computation," IBM Journal of Research and Development , vol.17, no.6, pp.525,532, Nov. 1973

doi: 10.1147/rd.176.0525

[3] B, Raghu Kanth; B, Murali Krishna; G, Phani Kumar; J, Poornima, "A Comparative Study of Reversible Logic Gates", InternationalJournal of VLSI & Signal Processing Applications, vol.2, Issue 1, Feb 2012, (51-55), ISSN 2231-3133 (Online).

[4] Morrison, M.; Ranganathan, N., "Design of a Reversible ALU Based on Novel Programmable Reversible Logic Gate Structures," VLSI (ISVLSI), 2011 IEEE Computer Society AnnualSymposium on , vol., no., pp.126,131, 4-6 July 2011

doi: 10.1109/ISVLSI.2011.30.

[5] Nachtigal, M.; Thapliyal, H.; Ranganathan, N., "Design of a reversible single precision floating point multiplier based on operand decomposition," Nanotechnology (IEEE-NANO), 2010 10thIEEE Conference on , vol., no., pp.233,237, 17-20 Aug. 2010

doi: 10.1109/NANO.2010.5697746 (Nachtigal, Thapliyal, &Ranganathan, 2010)

[6] John P. Hayes, "Computer Architecture and Organization", McGraw-Hill, 1998. ISBN 10: 0070273553 / ISBN 13: 9780070273559

[7] Min-Lun Chuang; Chun-Yao Wang, "Synthesis of Reversible Sequential Elements," Design Automation Conference, 2007. ASPDAC'07. Asia and South Pacific , vol., no., pp.420,425, 23-26 Jan. 2007 doi: 10.1109/ASPDAC.2007.358022

[8] Yelekar, P.R.; Chiwande, S.S., "Design of sequential circuit using reversible logic," Advances in Engineering, Science andManagement (ICAESM), 2012 International Conference on , vol., no., pp.321,326, 30-31 March 2012.