Indian Journal of Science and Technology, *Vol 7(4S), 34–39, April 2014*

ISSN (Print): 0974-6846
ISSN (Online) : 0974-5645

Pipelined Quadratic Equation based Novel Multiplication Method for Cryptographic Applications

B. Vignesh*, K. P. Sridhar and R. Parameshwaran

School of Computing, SASTRA University, India; vignesh9028@gmail.com, vkpsri@gmail.com, parameswaran@ict.sastra.edu

Abstract

Digital number system in a computer performs the basic arithmetic operation using multiplier. The functionality of this multiplier can be obtained through addition, subtraction and right shift operations but the utilization of adders, subtractors and shift registers components to perform multiplication is huge. So we proposed the multiplication algorithm for computer by altering the existing booth methodology, which reduces the component and complexity of the multiplication process. The novel algorithm is implemented on pipeline quadratic equations and verified through the simulation result. The arithmetic circuit is written in VHDL and synthesized with Xilinx using Spartan 3 FPGA kit.

Keywords: Computer Arithmetic Multiplication Algorithm, FPGA, Quadratic Equation, VLSI

1. Introduction

Computer electronic number system has digital arithmetic circuit to provide the multiplication, division, and exponentiation operations. All this function uses a multiplier as the basic building blocks. To increase the performance of computation arithmetic circuit, the multiplication process is tackle through Robertson and Booth algorithm which uses right shift and addition or subtraction function to perform multiplication in faster manner. Hence complex arithmetic digital circuit in Very Large Scale Integration (VLSI) is handling easily through the existing algorithm. To make the computation even faster, the conventional multiplication algorithm is proposed by modifying the booth algorithm.

In general there are different types of number system for computer operation based on their hardware specifications. The radix number and signed number representation are used in our multiplication process. Both the number representation takes place as fixed and floating point number systems. The radix number system is represented through their weights along with the digits (D.R), where D is the digit and R is the radix. Radix indicates the types of digit which represent the number system. If R is equal to 10 it is a decimal number system or 2 it is a binary number system. Example: $(20.0)_{10} = 2 \times 10^1 + 0 \times 10^0 + 0 \times 10^{-1}$, which shows that above example is decimal number system. The dot point within the digit is called as radix point. The digits before the radix point are indicated through the multiplication of 10 to the power based on their position and similarly the digits after the radix point are indicated through division of 10 to the power based on their positions. For a signed value representation, the digit on the most significant bit position shows the type of signed value. If the bit is 1, then the digit represents a negative signed value. If the bit is 0, then the digit represents a positive signed value. For example: $(1\ 1\ 0\ 0\ 1)_2 = (-9)_{10}$ shows the negative signed value. Table 1 shows different representation of signed number system. Notice that, if the most significant digit represented by 0, then their complement value is same as signed value. For the negative signed value the complement value is same as general

*Author for correspondence

Table 1. Complement value for signed number

Binary Values	Signed value	Complement value	
		1's	2's
0001	1	1	1
0011	3	3	3
1001	–1	–6	–7
1011	–3	–4	–5

complement not equal to signed value. The complement value is necessary for performing subtraction during the computer multiplication process.

All the above representation is fixed point representation. For the floating point value it represents the radix value presented along with the exponent value. Example $D \times r^e$ where e is exponent, r is as usual radix and D is the significant digit.

The rest of the paper is organized as follow. The section 2 describes about the existing process system. The different multiplication process along with proposed method is explained in section 3. In section 4, hardware implementation for quadratic pipeline system along with simulation and synthesis report is presented. Finally we conclude our entire paper on section 5.

2. Existing Method

Redundant method[1] of arithmetic operation is introduced to perform addition and multiplication for cryptographic application. Operation is based on their redundant bit and carry bit which reduces the storage of product for each multiplier. To reduce the area and power consumption in the chip a traditional method of Tree Addition and Vedi Multiplication is introduced in paper[2]. The process handles through accumuting and multiplying the desire function values. Matrix multiplication[3] for crypto processor is introduced which performs the matrix multiplication in series, concurrent and parallel manner from one core to other core at simultaneous clock cycle which reduces the latency of the arithmetic digital circuit.

The Vedic Multiplier[4] using Urdhva Tiryagbhyam algorithm is proposed which perform the multiplication and accumulation operation using adder and accumulator. The speed of vedic circuit is high due to carry propagation techniques. To reduce the critical path and latency in the circuit, linear array of systolic is proposed in paper[5], by adding the longest carry save adder into the circuit we can achieve high speed and reduces the distortion in amplification during broadcasting. A survey of different algorithm for multiplication[6–8] is presented. All these algorithm are complex in nature hence it very difficult to implement. So this paper proposes the novel method of multiplication with high performance and less complexity.

3. Proposed Method

3.1 Algorithm for Multiplication

Digital Multiplication plays the vital role in processor at computer electronic system. The normal multiplication involves shift and add operation to determine the product term. Each term in a multiplier is multiplied with the multiplicand to obtain the product term and product term is ordered one by one with towards the left shift. Finally shifted product term is added to determine the final product. On system point of perspective it is impossible to store all the product terms from multiplier. Hence novel machine level multiplication method is introduced with right shift operation. Overall adder and subtractor are used for multiplication process in machine level is reduced in this novel method. Let us consider the two signed binary value $X = (1\ 1\ 0\ 0\ 1)_2$ and $Y = (1\ 0\ 1\ 1\ 1)_2$ where the most significant bit of both binary values represent the signed value. The negative signed value is indicated by 1 and positive signed is indicate value by 0. Let see the performance for different types of multiplication operation.

3.1.1 Normal Method

Multiplicand $X = (1\ 1\ 0\ 0\ 1)_2 = (-9)_{10}$

Multiplier $Y = (1\ 0\ 1\ 1\ 1)_2 = (-7)_{10}$

$$
\begin{array}{r}
11001 \times 10111 \\
\hline
11001 \\
11001 \\
11001 \\
00000 \\
11001 \\
\hline
(1\ 1\ 1\ 1\ 1\ 1)_2 = (63)_{10}
\end{array}
$$

3.1.2 Robertson Method

In Robertson method, four different types of methodologies are followed based on their signed value of X and Y.

If both X and Y are positive, the multiplication operation is taking place as a series of right shift and add operation. The operation is decided based on the LSB bit of multiplier. If LSB = 1, ADD and SHIFT operation take place in the round. If LSB = 0, only SHIFT operation will take place. Hence continuous right shift is obtained through adding 0 at MSB in each shift. The total number of round is determined through number of bits in multiplier and multiplicand. The final product term obtained is positive. If X is positive and Y is negative or vice-versa, the final product obtained is negative. The subtraction is taken place in last round instead of addition. If X and Y are negative, the final product obtained is positive but subtraction will take place at last round. Table 2 shows the fourth case of ROBERTSON method which computes the multiplication process.

3.1.3 Booth Method

In booth method the current operation is identified from last two bits of multiplier. In default, 0 is added to the LSB of multiplier for the first round. Based on the last two bits of multiplier shift and operations will take place at each round. If the last two bits are either 00 or 11 only right shift will take place. If last two bits 01, addition of X and then shifting will take place. If last two bits 10, subtraction of X and then shifting will take place. The total number of rounds is same as Robertson method. In this method addition and subtraction is occupied randomly not like a complete addition as Robertson method. During shifting leading of 0's or 1's at MSB is based on their right most neighbour bit in the strings.

The Table 3 shows Booth method of computation. The bottleneck of this method is consumption large number of adder and subtraction.

3.1.4 Novel Multiplcation Algorithm from Altered Booth Method

The booth method is enhanced through reducing the addition and subtraction process based on the consecutive 0's and 1's in multiplier and multiplicand. The operation determine from last three bits in multiplier. The total number of round involved in this algorithm is half of the above two method. The functionality of operations based of last three bits shown in Table 4. In default, extra bit 0 is added next to LSB of multiplier for first round. The total number of rounds is calculated through diving the total bit in multiplier. The resultant whole value is total number of rounds in novel method. Hence the computation time is also reduced in proposed method.

Table 2. Robertson Method of Computation

Round	Computation	0 0 0 0 0	1 0 1 1 1
1	LSB bit of Multiplier = 1 then Add X and do right shift	1 1 0 0 1 1 1 0 0 1 1 1 1 0 0	1 0 1 1 1 1 1 0 1 1
2	LSB bit of Multiplier = 1 then Add X and do right shift	1 1 0 0 1 1 0 1 0 1 1 1 0 1 0	1 1 0 1 1 1 1 1 0 1
3	LSB bit of Multiplier = 1 then Add X and do right shift	1 1 0 0 1 1 0 0 1 1 1 1 0 0 1	1 1 1 0 1 1 1 1 1 0
4	LSB bit of Multiplier = 0 then do right shift	1 1 1 0 0	1 1 1 1 1
5	LSB bit of Multiplier = 1 then Sub X and do right shift	0 0 1 1 1 0 0 0 1 1 0 0 0 0 1	1 1 1 1 1 1 1 1 1 1

$(0\,0\,0\,0\,1\,1\,1\,1\,1\,1)2 = (63)_{10}$

Note: Subtraction is carried out through taking 2's complement for X and adding the desire result with their operation at specific place

Table 3. Booth method of computation

Round	Computation	0 0 0 0 0	1 0 1 1 1	0
1	Last 2 bits of Multiplier = 10, Sub X and do right shift	0 0 1 1 1 0 0 1 1 1 0 0 0 1 1	1 0 1 1 1 1 1 0 1 1	1
2	Last 2 bits of Multiplier = 11, do right shift	0 0 0 0 1	1 1 1 0 1	1
3	Last 2 bits of Multiplier = 11, do right shift	0 0 0 0 0	1 1 1 1 0	1
4	Last 2 bits of Multiplier = 01, Add X and do right shift	1 1 0 0 1 1 1 0 0 1 1 1 1 0 0	1 1 1 1 0 1 1 1 1 1	0
5	Last 2 bits of Multiplier = 10, Sub X and do right shift	0 0 1 1 1 0 0 0 1 1 0 0 0 0 1	1 1 1 1 1 1 1 1 1 1	1

$(0\,0\,0\,0\,1\,1\,1\,1\,1\,1)2 = (63)_{10}$

Table 5 shows that multiplication process through novel method which involves 3 rounds and requiring only three operations for computation but the both ROBERSTON and BOOTH methodology requires more than four numbers of addition or subtraction operations for multiplication process.

3.2 Quadratic Function using Pipeline Architecture

The polynomial function having maximum three coefficients is known as quadratic polynomial. The below architecture shows the pipelined quadratic polynomial which perform multiplication based of novel multiplication process with less number of addition and subtraction. The relation of quadratic polynomial is given by $Ax^2 + Bx + C$.

This Quadratic function has the input of 8 bit signed value and 8 bit coefficients value which will generate the output of 24 bit fixed point signed value. Coefficient is varied dynamically based on their clock cycle.

Overall latency of the output is three cycles of clock. The computation of operations function did not need any intermediate storage values to generate result. The Figure 1 shows the pipeline architecture of Quadratic function.

Table 4. Operation for Proposed Method

Last Three Bits	Operation needs to Perform
0 0 0	Multiple Right Shifts
0 0 1	Add X then Multiple Right Shifts
0 1 0	Add X then Multiple Right Shifts
0 1 1	Single right shift, Add X then again Right Shift
1 0 0	Single right shift, Add X then again Right Shift
1 0 1	Sub X then Multiple Right Shifts
1 1 0	Sub X then Multiple Right Shifts
1 1 1	Multiple Right Shifts

Table 5. Proposed Method of Computation

Round	Computation	0 0 0 0 0	1 0 1 1 1	0
1	Last 3 bits of Multiplier = 110 then Sub X and right shift twice	0 0 1 1 1 0 0 1 1 1 0 0 0 1 1 0 0 0 0 1	 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1	 1 1
2	Last 3 bits of Multiplier = 011 then Shift Right, Add X and again right shift	0 0 0 0 0 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0	1 1 1 1 0 1 1 1 1 0 1 1 1 1 1	1 0
3	Last 3 bits of Multiplier = 110 then Sub X and right shift twice	0 0 1 1 1 0 0 0 1 1 0 0 0 0 1	 1 1 1 1 1 1 1 1 1 1	 1

4. Experimental Result

The hardware implementation of pipelined quadratic architecture is shown in Figure 1. This electronic circuit is described through Very High Speed Hardware Description Language (VHDL) and simulated on MODELSIM. The circuit has the input of three 8 bit co efficient fractions and another 8 bit signed input. There are 15 different signals which are used to produce the output of 24 bit signed fraction value. The system is activated through enabling en value to 1.

The simulation result of pipelined quadratic test bench is shown in Figure 2 which runs until 2100 ns. The synthesis report is generated on Xilinx ISE for Spartans 3 Field Programmable Gated Array (FPGA) kit. The Pipelined quadratic is used to determine the higher polynomial value at different curve on graph. The continuous operations will provide higher polynomial value of input for example cubic value of x^3. We can also able to estimate the trigonometric values such as Sin, Cos, Tan. The logarithm and exponential values can also be determined through series of multiplication operation. Matching the look up table values with the bounded values is also possible with the Quadratic polynomial.

O - Optimized (High Performance),

X - Non Optimized (Low Performance).

3.3 Applying Novel Multiplication Method on Cryptographic Application AES (Advance Encryption Standard)

The proposed novel method of multiplication is applied on AES cryptographic algorithm during the columns mix operation. The AES should generate the cipher text through performing certain number of operation like Substitution box, Mix Columns, Adding Round keys. During Mix Columns it performs finite field multiplication using Quadratic equation. The proposed multiplication was applied on this Quadratic equation and simulation result shown on Figure 2.

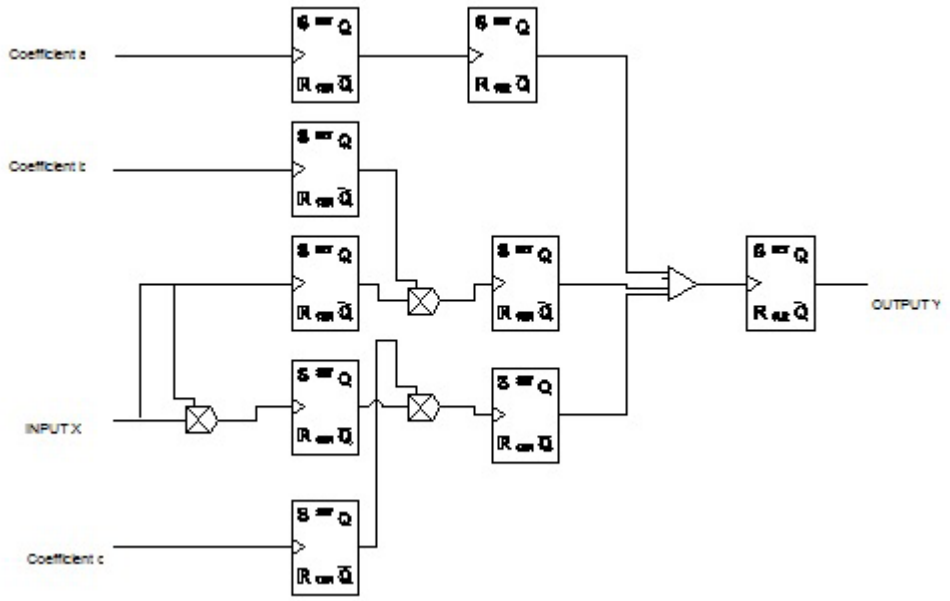

Figure 1. Pipeline architecture of Quadratic function.

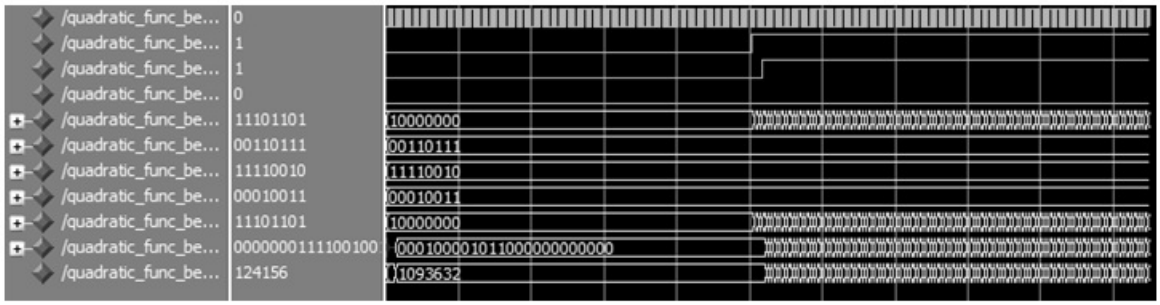

Figure 2. Simulation result for pipelined quadratic equation.

The device utilization for the implemented quadratic pipeline function is shown on Table 6. Table 7, shows the comparison of normal and proposed multiplication method in case of operations, speed and delay of output due to latency involved in the digital circuit. Comparison between throughput frequencies is also done with the synthesis report generated.

4.1 Applications of Novel Multiplication Method

A symmetrical multi crypto processor[9] is designed through complex Montgomery modified algorithm and modular arithmetic. The multiplication is implemented by modified algorithm. The similar multiplication process

Table 6. Device Utilization for Pipelined Quadratic Equation $Ax^2 + Bx + C$

FPGA DEVICE UTILIZATION **Device Selected : 3s50pq208-5 (SPARTAN 3)**	
Total number of Slices	19
Total number of 4 input Look Up Tables	35
Total number of flip flop Slices	32
Total number of Shift registers	8
Minimum time period	5.613
Frequency Maximum	178.148

Table 7. Comparison of normal and proposed method

Multiplication Technique	Operations	Speed	Latency	Throughput
Proposed	O	O	X	O
Normal	X	X	O	X

can also be obtained through novel altered booth method. Hence our proposed algorithm is used to implement cryptographic multiprocessor which also reduces the complexity of circuit. The different types of cryptographic techniques[10] are generated through Kasumi and Seed algorithm for multiplication. These cryptographic techniques can also be generated through the proposed multiplication method. The novel multiplication process is also applicable for other general application such as

i) Symmetric key cryptographic algorithm to find the exponential values.
ii) High level processor.
iii) Slope value estimation in graphical Curve.
iv) Trigonometric Value calculation.
v) CORDIC Algorithm

5. Conclusion

This paper proposed the novel method of computer multiplication with less complexity. The current method is obtained through altering the booth algorithm. The proposed algorithm is implemented on pipeline quadratic function to evaluate the functionality and it is verified through the simulation result. The synthesis report shows the reduction of component, area and latency on arithmetic digital circuit.

6. References

1. Kawakami K, Shigemot K, Nakano K. Redundant radix-number system for accelerating arithmetic operations on the FPGAs. Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies; 2008 Dec 1–4; Otago. IEEE; p. 370–77.
2. Mishra N, Malviya U. Efficient area and speed optimized multiplication technique using vedic and tree addition structure. ACSIJ Advances in Computer Science: an International Journal. 2013; 2(3):42–47.
3. Ismail MA, Mirza SH, Altaf T. Concurrent matrix multiplication on multi-core processors. Int J Comput Sci Secur. 2011; 5(2):208–20.
4. Pradhan M, Panda R, Sahu SK. MAC implementation using vedic multiplication algorithm. Int J Comput Appl. 2011; 26–28.
5. Liu J, Dong J. Design and implementation of an efficient montgomery modular multiplier with a new linear systolic array. IEEE International Conference on Information Theory and Information Security (ICITIS); 2010 Dec 17–19; Beijing. IEEE. 225–29.
6. Nedjah N, Mourelle LD. A review of modular multiplication methods and respective hardware implementations. Informatica(Slovenia). 2006; 30(1):111–129.
7. Swartzlander EE. The Quasi-serial multiplier. IEEE Trans Comput. 1973; 317–21.
8. Korneichuk VI, Shlem BM. Analysis of algorithms of fast multiplication in digital computers. Cybernetics. 1974 May–Jun; 446–49.
9. Shuguo L, Runde Z, Yuanqing G. A 1024-bit RSA crypto-coprocessor for smart cards, 4th International Conference on ASIC Proceedings, ASICON; 2001; Shanghai. IEEE. p. 352–55.
10. Kim H, Lee S. Design and implementation of a private and public key crypto processor and its application to a security system. IEEE Trans Consum Electron. 2004; 50(1):214–24.