Journal of Computer Science 5 (1): 49-56, 2009
ISSN 1549-3636

Efficient Reversible Montgomery Multiplier and Its Application to Hardware Cryptography

Noor Muhammed Nayeem, Lafifa Jamal and Hafiz Md. Hasan Babu
Department of Computer Science and Engineering, University of Dhaka, Bangladesh

Abstract: Problem Statement: Arithmetic Logic Unit (ALU) of a crypto-processor and microchips leak information through power consumption. Although the cryptographic protocols are secured against mathematical attacks, the attackers can break the encryption by measuring the energy consumption. **Approach:** To thwart attacks, this study proposed the use of reversible logic for designing the ALU of a crypto-processor. Ideally, reversible circuits do not dissipate any energy. If reversible circuits are used, then the attacker would not be able to analyze the power consumption. In order to design the reversible ALU of a crypto-processor, reversible Carry Save Adder (CSA) using Modified TSG (MTSG) gates and architecture of Montgomery multiplier were proposed. For reversible implementation of Montgomery multiplier, efficient reversible multiplexers and sequential circuits such as reversible registers and shift registers were presented. **Results:** This study showed that modified designs perform better than the existing ones in terms of number of gates, number of garbage outputs and quantum cost. Lower bounds of the proposed designs were established by providing relevant theorems and lemmas. **Conclusion**: The application of reversible circuit is suitable to the field of hardware cryptography.

Key words: Montgomery multiplier, carry save adder, reversible gate, shift register

INTRODUCTION

Power analysis is a physical attack to cryptosystems such as smart card, tamperproof "black box" and microchip. It exploits the fact that the power dissipation of an electronic circuit depends on the actions performed in it. Kocher *et al*.[1] describe Simple Power Analysis (SPA) and Differential Power Analysis (DPA) attacks which use the power-dissipation characteristics as a provider of side-channel information. Using DPA, an attacker can extract information on secret keys by statistically analyzing power consumption measurements from multiple cryptographic operations performed by a crypto-processor.

DPA is more difficult to prevent, since even small biases in the power consumption can lead to exploitable weaknesses. In this study, the authors propose the use of reversible logic to protect the crypto-systems from power analysis attacks. According to Landauer[2,3], in logic computation every bit of information loss generates kTln2 joules of heat energy where k is Boltzmann's constant of 1.38×10^{-23} J/K and T is the absolute temperature of the environment. At room temperature the dissipating heat is around 2.9×10^{-21} J. Energy loss due to Landauer limit is also important as it is likely that the growth of heat generation causing information loss will be noticeable in future. Reversible circuits are fundamentally different from traditional irreversible one. In reversible logic, no information is lost, i.e., the circuit that does not lose information is reversible. Bennett[4] showed that zero energy dissipation would be possible if the network consists of reversible gates only. Thus the proposed reversible hardware will prevent any type of power analysis attack, since no energy will be dissipated from reversible circuits.

Modular multiplication is the most common operation in the cryptosystems, such as RSA, Elliptic Curve Cryptography (ECC), Digital Signature Algorithm (DSA) and Diffie-Hellman key exchange. It is also the critical part of the computing efficiency in the cryptosystem which involves modular multiplications with large integers for enhancing its security[7]. The most popularly used method for the fast implementations of modular multiplication is Montgomery's algorithm[8]. To avoid long carry propagation during the addition stages of the computation, several techniques such as systolic array and Carry Save Adder (CSA) architecture were found

Corresponding Author: Hafiz Md. Hasan Babu, Department of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh Tel: +880-1711-351055

in the literature[6,7]. This study focuses on the reversible CSA architecture implementation of Montgomery multiplier.

MATERIALS AND METHODS

Reversible gate: Reversible Gates are circuits in which the number of outputs is equal to the number of inputs and there is a one to one correspondence between the vector of inputs and outputs[9].

Let the input vector be I_v, output vector be O_v and they are defined as follows, $I_v = (I_i, I_{i+1}, I_{i+2} \ldots I_{k-1}, I_{k,})$ and $O_v = (O_i, O_{i+1}, O_{i+2 \ldots} O_{k-1}, O_k)$. For each particular i, there exits the relationship $I_v \leftrightarrow O_v$.

Garbage output: Unwanted or unused output of a reversible gate (or circuit) is known as Garbage Output.

Feynman gate (FG)[19] is used to perform Exclusive-OR between two inputs. But in that case, one extra output will be generated as well, which is the garbage output as shown in Fig. 1 with * .

Some major reversible gates required for this study are Fredkin gate (FRG)[10], Peres gate[11], TSG gate[12], modified TSG (MTSG) gate[13] and HNFG gate[20] which are shown in Fig. 2-6 respectively.

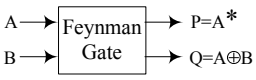

Fig. 1: Illustrating garbage output

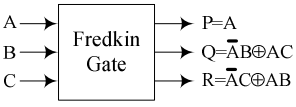

Fig. 2: Fredkin gate

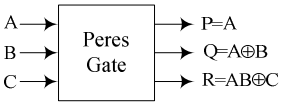

Fig. 3: Peres gate

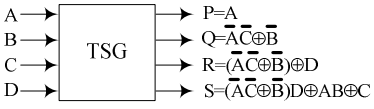

Fig. 4: TSG gate

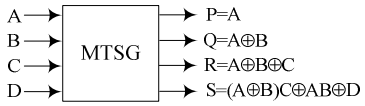

Fig. 5: MTSG gate

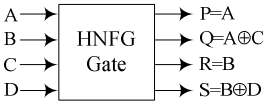

Fig. 6: HNFG gate

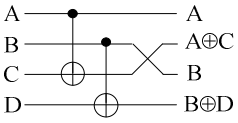

Fig. 7: Equivalent quantum representation of HNFG

Quantum cost: Every reversible gate can be calculated in terms of quantum cost and hence the reversible circuits can be measured in terms of quantum cost. The quantum cost of every 2×2 gate is the same and the cost is unity[5,15]. According to[5] a 1×1 gate costs nothing and every quantum gate can be realized from 1×1 and 2×2 gates and its cost calculated as a total sum of 2×2 gates used.

Reversible NOT gate has no quantum cost as it is a 1×1 gate. The quantum costs of Feynman, Fredkin, Peres, TSG and MTSG gates are 1, 5, 4, 13, 6 respectively[5, 13, 15]. Quantum equivalent representation of HNFG gate is shown in Fig. 7. The cost of the HNFG gate is two 2×2 gates, or simply 2.

Reversible logic in hardware cryptography: The main source of power consumption in hardware cryptography is the ALU of a crypto-processor. It consists of CSA, multipliers, registers, shift registers, accumulators and multiplexers. Therefore, the ALU of a crypto-processor can be designed using reversible logic so it will not dissipate any heat. Each component of the crypto-processor is described here.

Proposed reversible four-to-two CSA: TSG gate is very popular to construct the full adder circuit. But TSG is very complex in nature and its quantum cost is extremely high which is 13. MTSG is very useful to realize full adder as its quantum cost is very low (only 6) as compared to the TSG. Realization of full adder

using MTSG is shown in Fig. 8. Using MTSG gates Fig. 9 shows the proposed four-to-two reversible CSA.

Table 1 shows that the proposed reversible CSA requires lower quantum cost than the existing one found in the literature[16].

Proposed reversible register: Figure 10 shows the implementation of reversible clocked D flip-flop[16,17]. The reversible D flip-flops can be used to implement a reversible register. The proposed n-bit reversible register is shown in Fig. 11.

Theorem 1: An n-bit reversible register can be realized by at least 2n gates and n+1 garbage outputs.

Proof: An n-bit reversible register is designed using n reversible clocked D flip-flops. From[16,17], each reversible D flip-flop contains one Fredkin gate and one Feynman gate, a total of two gates and produces two garbage outputs. In reversible register, CLK output of a Fredkin gate is connected to the CLK input of the Fredkin gate of next D flip-flop. Thus, reversible register reduces one garbage output from each D flip-flop except the last one. Therefore, an n-bit reversible register can be realized by at least 2n gates and n+1 garbage outputs.

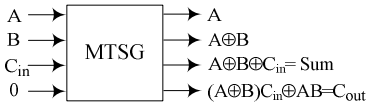

Fig. 8: Reversible clocked D flip-flop

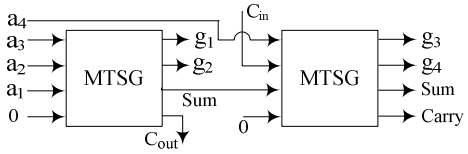

Fig. 9: Proposed four-to-two reversible CSA using MTSG gates

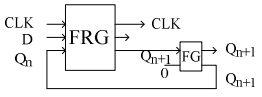

Fig. 10: Reversible clocked D flip-flop

Table 1: Comparison of different reversible CSAs

	Quantum cost
Existing Circuit using TSG[16]	13×2 = 26
Proposed Circuit using MTSG	6×2 = 12

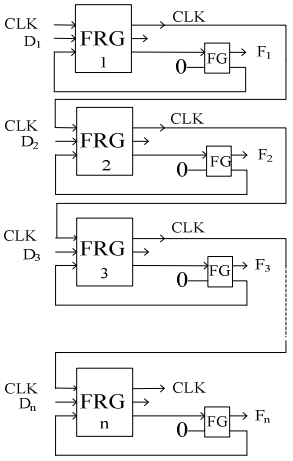

Fig. 11: Proposed n-bit reversible register

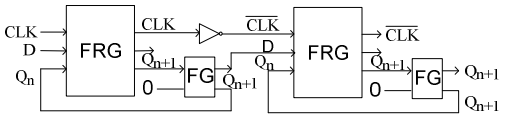

Fig. 12: Reversible master-slave D flip-flop

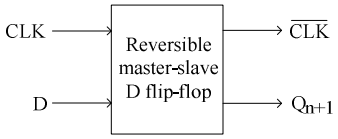

Fig 13: Block diagram of reversible master-slave D flip-flop

Table 2: Comparison of different n-bit reversible registers

	No. of gates	Garbage output	Quantum cost
Existing circuit[16]a	2n	2n	6n
Proposed circuit	2n	n+1	6n

[a]: The design in[16] contains multiple fan-outs, which are forbidden in strict reversible sense

Lemma 2: The quantum cost of an n-bit reversible register is at least 6n.

Proof: From[5,15], quantum costs of Feynman gate and Fredkin gate are one and five respectively. Since from[16,17], each reversible D flip-flop contains one Fredkin gate and one Feynman gate, the quantum cost of D flip-flop is 1+5 = 6. Since there are n D flip-flops, the quantum cost of an n-bit reversible register is 6n.

Comparative results of different reversible n-bit registers are shown in Table 2.

Proposed reversible shift register: Design of reversible master-slave D flip-flop[17] and its block diagram are shown in Fig. 12 and 13 respectively.

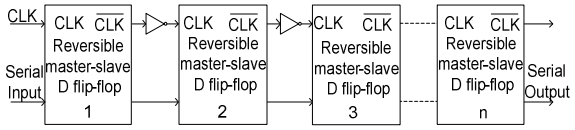

Fig. 14: Proposed reversible n-bit shift register

Figure 14 shows the proposed n-bit reversible shift register built from reversible master-slave D flip-flops. Each clock pulse shifts the contents of the register one bit position to the right. This shift register can be classified as a Serial-In, Serial-Out (SISO) shift register.

Theorem 3: An n-bit reversible SISO shift register using master-salve D flip-flops can be realized by at least 6n-1 gates and 2n+1 garbage outputs.

Proof: An n-bit reversible SISO shift register can be designed using n reversible master-slave D flip-flops. From[17], each reversible master-slave D flip-flop contains two Fredkin gates, two Feynman gates and one reversible NOT gate, a total of five gates and produces three garbage outputs. In reversible shift register, $\overline{CLK}$ output of a D flip-flop (2nd Fredkin gate) is connected to a reversible NOT gate and the inverted output is connected to CLK input of the next D flip-flop (1st Fredkin gate of next D flip-flop). Therefore, reversible shift register reduces one garbage output from each D flip-flop except the last one, i.e. it produces 2(n-1) garbage outputs for first n-1 D flip-flops and 3 garbage outputs for last D flip-flop. So, total garbage output is 2(n-1) + 3 = 2n+1.

The n-bit reversible shift register requires 5n gates for n master-salve D flip-flops and n-1 reversible NOT gates, a total of 6n-1 gates. Therefore, an n-bit reversible shift register can be realized by at least 6n-1 gates and 2n+1 garbage outputs.

Lemma 4: The quantum cost of an n-bit reversible SISO shift register using master-slave D flip-flops is at least 12n.

Proof: From[5,15] quantum costs of reversible NOT gate, Feynman gate and Fredkin gate are zero, one and five respectively. From[17], each reversible master-slave D flip-flop contains two Fredkin gates, two Feynman gates and one reversible NOT gate. So, the quantum cost of master-slave D flip-flop is 5×2+1×2+0×1 = 12. Since an n-bit reversible SISO shift register contains n master-salve D flip-flops, the quantum cost is 12n.

Table 3: Comparison of different reversible n-bit shift registers

	No. of gates	Garbage output	Quantum cost
Existing circuit[16]	6n	4n+1	14n
Proposed circuit	6n-1	2n+1	12n

Table 4: Truth table for the proposed reversible PIPO shift register

HOLD	E	Final Output Q_i^+
0	0	Q_{i-1} (Right shift)
0	1	I_i (Parallel load)
1	Don't care	Q_i (No change)

The optimality of the proposed design can be easily understood from Table 3 which shows the comparative study using existing reversible shift register. The proposed design is less costly in terms of number of gates, garbage bits and quantum costs than the existing one

Proposed reversible Parallel-In, Parallel-Out (PIPO) shift register using clocked D flip-flops: In PIPO shift register, all data bits are loaded into the register at once with the next clock pulse. After shift operation all data bits appear on the parallel outputs immediately. The control inputs (HOLD, E) select the operation of the register according to the function entries in Table 4.

From Table 4, when both HOLD and E are low, the shift register performs the shift-right operation. When HOLD is low and E is high, the inputs I_1, I_2, …, I_n are loaded in parallel into the register coincident with the next clock pulse. The outputs O_1, O_2, O_3, …, O_n are available in parallel from the Q output of the flip-flops. When HOLD is high, present value of flip-flop is applied to the D input of that flip-flop. In other words, the register is inactive when HOLD is high and the contents are stored indefinitely.

The characteristic function of Q_i^+ can be obtained from Table 4:

$$Q_i^+ = \overline{HOLD}.E.I_i + \overline{HOLD}.\overline{E}.Q_{i-1} + HOLD.Q_i \quad (1)$$

For the first stage Q_{i-1} is the serial input (SI) and for the last stage Q_i is the serial output (SO).

Theorem 5: The characteristic function of Q_i^+ of reversible PIPO shift register (using clocked D flip-flop) can be obtained by 2 gates and 4 garbage outputs with 10 quantum cost.

Proof: The characteristic function of Q_i^+ is:

$$Q_i^+ = \overline{HOLD}.E.I_i + \overline{HOLD}.\overline{E}.Q_{i-1} + HOLD.Q_i$$
$$= \overline{HOLD}.(E.I_i + \overline{E}.Q_{i-1}) + HOLD.Q_i$$

Thus, this function can be implemented by only two Fredkin gates as shown in Fig. 15. From Fig. 15, it is clear that it generates 4 garbage outputs. As the quantum cost[15] of a Fredkin gate is 5, the implementation of characteristic function of Q_i^+ requires 10 quantum cost.

Figure 16 shows the basic cell of the proposed PIPO shift register using clocked D flip-flop and its block diagram. By cascading n basic cells, PIPO shift register can be implemented as shown in Fig. 17.

Theorem 6: The n-bit reversible PIPO shift register using clocked D flip-flops can be implemented by 5n reversible gates and 3n+3 garbage outputs.

Proof: Each basic cell has two parts: generation of Q_i^+ and D flip-flop. From Theorem 5, generation of Q_i^+ requires 2 gates and produces 4 garbage outputs. But in n-bit PIPO shift register, HOLD and E outputs of each basic cell are connected to the HOLD and E inputs of the next basic cell respectively. Thus it reduces 2 garbage outputs from each basic cell in the generation of Q_i^+ except the last cell. In other words, generation of Q_i^+ ($1 \le i < n$) produces only 2 garbage outputs and Q_n^+ (that is SO) produces 4 garbage outputs.

In the basic cell, each D flip-flop (Fig. 16) requires 3 gates and has four Q_i outputs. One of the Q_i outputs is used to the generation of Q_i^+, second one is used to generate Q_{i+1}^+, one is considered as parallel output and other is fed to its own D flip-flop's Fredkin gate. CLK output of each basic cell is connected to the CLK input of the next basic cell.

Thus each of the basic cells (except the last cell) produces only 1 garbage output from each D flip-flop. D flip-flop in last cell produces 2 garbage outputs. Thus, each basic cell requires 2+3 = 5 gates. Each of the first n-1 basic cells produces 2+1 = 3 garbage and the last cell produces 4+2 = 6 garbage outputs. As an n-bit PIPO shift register has n basic cells, it requires 5n gates and produces (n-1)×3+6 = 3n+3 garbage outputs.

Lemma 7: The quantum cost of an n-bit reversible PIPO shift register using D flip-flops is at least 18n.

Proof: From theorem 5, generation of Q_i^+ requires quantum cost of 10. From[5,15] quantum costs of Feynman gate and Fredkin gate are one, five respectively. From Fig. 6 quantum cost of HNFG gate is two. D flip-flop of each basic cell requires one Fredkin gate, one Feynman gate and one HNFG gate, a total quantum cost of 5+1+2 = 8. Thus, an n-bit reversible PIPO shift register requires 10×n+8×n = 18n quantum cost.

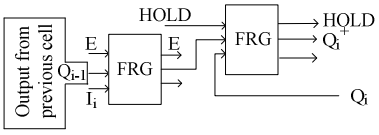

Fig. 15: Implementation of characteristic function of Eq. 1

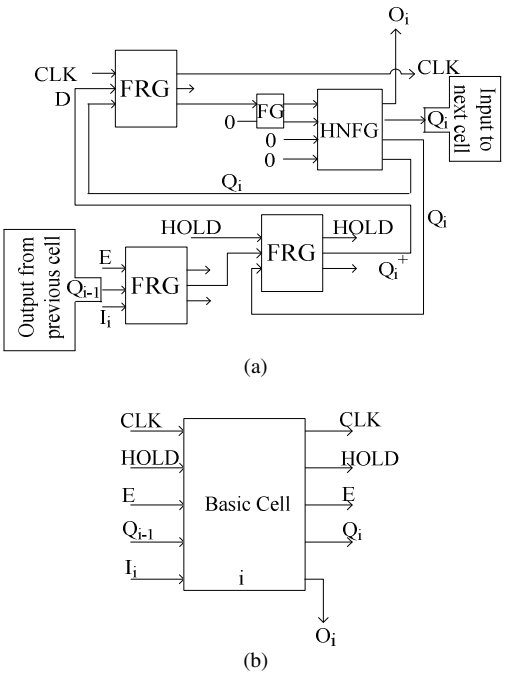

Fig. 16: Basic cell for the proposed reversible PIPO shift register: (a) Structure; (b) Block diagram

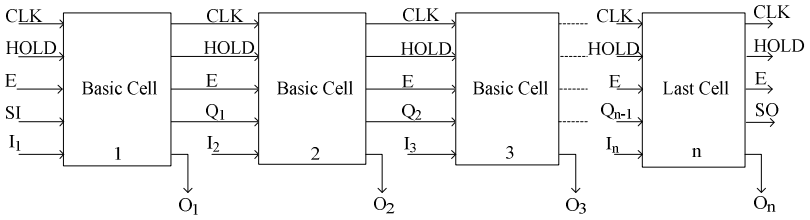

Fig. 17: Proposed reversible PIPO shift register using D flip-flops

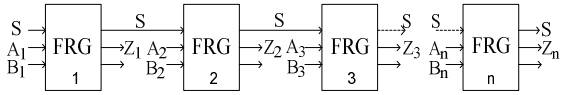

Fig. 18: Proposed 2-input n-bit reversible multiplexer

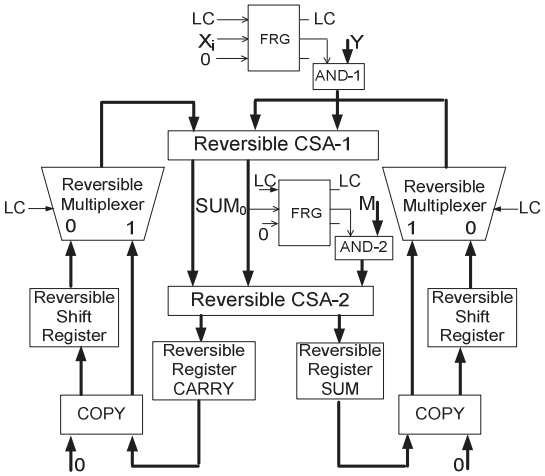

Fig. 19: Proposed architecture of reversible Montgomery multiplier

Proposed 2-input reversible multiplexer: As Fredkin gate is a controlled swap gate, multiplexer can be implemented using this gate. Figure 18 shows a proposed 2-input n-bit reversible multiplexer where S is the select input, $A_1A_2A_3\ldots A_n$ and $B_1B_2B_3\ldots B_n$ are two inputs. If S = 0, then $(Z_1Z_2Z_3\ldots Z_n) = (A_1A_2A_3\ldots A_n)$ or if S = 1, then $(Z_1Z_2Z_3\ldots Z_n) = (B_1B_2B_3\ldots B_n)$.

Therefore, a 2-input n-bit reversible multiplexer requires n Fredkin gates and produces n garbage outputs.

Proposed architecture of reversible montgomery modular multiplier: In hardware cryptosystems, the Montgomery multiplication algorithm[7,8] is used for modulo multiplication. The modified algorithm presented in[7] is very efficient. The authors make this algorithm compatible for reversible circuit and improve the efficiency of the proposed architecture.

Figure 19 shows the reversible implementation of the Modular Multiplier using the reversible components shown in this study. In the Fig. 19, darker lines represent multiple bits whereas lighter lines represent single bit. The AND-1 block performs the AND operation with X_i and Y which can be implemented using Peres gates. The AND-2 block performs the AND operation with SUM_0 and M. To get another copy of SUM_0, one Feynman gate cam be used but for simplicity this is not shown in Fig. 19. The COPY blocks are used to copy the signals to avoid the fan-out problems. These can be implemented using Feynman gates.

The working principle of this proposed reversible architecture is described in Algorithm 1. After the execution, the SUM register stores the result $XY\times2^{-(n+2)}$ mod M.

Algorithm 1 (Montgomery modular multiplication):

Inputs: X,Y,M with $0 \le X, Y < 2M$ and $2^{n-1} < M < 2^n$
Output: $X\times Y\times2^{-(n+2)}$ mod M
X_i: i^{th} bit of X
SUM_0: LSB of SUM
MMM (X, Y, M)
Step 1: Initialize SUM and CARRY to 0
Step 2: Set i := 0
Step 3: Set LC := 0
Step 4: Repeat Steps 5 to 9 while $i \le n + 1$
Step 5: Perform carry save addition on SUM, CARRY and $X_i\times Y$ using reversible CSA-1
Step 6: The outputs (SUM, CARRY) from reversible CSA-1 are fed into reversible CSA-2. Add these outputs with $SUM_0\times M$ using reversible CSA-2
Step 7: Store the results of above Step into reversible registers (SUM and CARRY)
Step 8: Perform right shift operation on SUM and CARRY by 1 bit using reversible shift registers. The results (SUM and CARRY) of these operations are fed back into the inputs of CSA-1
Step 9: Set i := i + 1
Step 10: Set LC := 1
Step 11: Repeat Steps 12 and 13 while CARRY ≠ 0
Step 12: Perform carry save addition on SUM and CARRY using reversible CSA-1 and CSA-2
Step 13: Store the results of above Step into reversible registers (SUM and CARRY)
Step 14: Return SUM = $X\times Y\times2^{-(n+2)}$ mod M

RESULTS AND DISCUSSION

Evaluation of the proposed architecture of reversible montgomery multiplier: Advantages of the proposed reversible architecture over existing reversible one[16] are as follows:

- Proposed architecture does not require any carry propagation logic (reversible ripple carry adder or carry look-ahead adder). Thus fast summation is possible.

- As it reuses the reversible CSA architecture to perform the full addition, it reduces area required for the circuit.
- This architecture is simple and efficient as it requires only two reversible CSAs.
- Unlike[16], this multiplier does not require to perform any subtraction operation.

Theorem 8: The proposed n-bit reversible Montgomery multiplier can be realized by at least 22n+25 gates and 16n+28 garbage outputs with quantum cost of 80n+91.

Proof: Since $2^{n-1}<M<2^n$, M is n bits long and since $0\leq X, Y<2M$, we assume that X and Y are n+1 bits long.

The proposed Reversible Montgomery Multiplier contains:

- Four-to-two n+1 CSAs, requiring 2(n+1) MTSG gates, 4(n+1) garbage outputs, 12(n+1) quantum cost
- 2(n+1) Peres gates to compute AND-1 and AND-2 blocks, producing 2(n+1) garbage outputs, 8(n+1) quantum cost
- Two (n+1)-bit reversible multiplexers, requiring 2(n+1) Fredkin gates, 2(n+1) garbage outputs, 10(n+1) quantum cost
- Two (n+1)-bit reversible PIPO shift registers, requiring 10(n+1) gates, 2(3n+6) garbage outputs, 36(n+1) quantum cost
- Two (n+1)-bit registers, requiring 2(2n+2) gates, 2(n+2) garbage outputs, 2(6n+6) quantum cost
- Two Fredkin gates for selections, producing 4 garbage outputs with 10 quantum cost
- 2(n+1) Feynman gates to construct two COPY blocks, requiring quantum cost of 2(n+1)
- One Feynman gate to get another copy of SUM_0 which requires one quantum cost

Therefore, the proposed architecture needs 2(n+1) +2(n+1)+2(n+1)+10(n+1)+2(2n+2)+2+2(n+1)+1 = 22n +25 reversible gates and produces 4(n+1)+2(n+1) +2(n+1)+2(3n+6)+2(n+2)+4 =16n+28 garbage outputs with quantum cost of 12(n+1)+8(n+1)+10(n+1) +36(n+1)+2(6n+6)+10+2(n+1)+1= 80n+91.

Applications of reversible montgomery multiplier: Montgomery multiplication is used in modular arithmetic as an efficient way of performing an exponentiation of two numbers modulo a large number, that is, A^B mod N. Algorithm 2[18] demonstrates the computation of A^B mod N used in RSA encryption and decryption functions.

Algorithm 2 (Efficient Algorithm to compute A^B mod N):

Inputs: A, B(exponent) = $(1b_{k-2}b_{k-3}, .. b_2b_1b_0)_2$, N(modulus), C(constant) = $2^{(n+2)}$ mod N
Output: R = A^B mod N, 0 <= R < N
MME (A, B, N, C)
Step 1: Set A' := MMM (A, C, N)
Step 2: Set R := A'
Step 3: Set i := k - 2
Step 4: Repeat Steps 5 to 7 while $i \geq 0$
Step 5: Set R := MMM(R, R, N)
Step 6: If ($b_i = 1$) Then Set R := MMM (R, A', N)
Step 7: Set i := i - 1
Step 8: Set R := MMM(R, 1, N)
Step 9: Return R

Algorithm 2 demonstrates the use of reversible architecture of Montgomery multiplier. Therefore this reversible architecture can be used in RSA, DSA, Diffie-Hellman key exchange and ECC crypto-processors to thwart DPA attacks.

CONCLUSION

In this study, reversible logic syntheses were carried out for the primitive components of the ALU of a reversible crypto-processor. The proposed reversible design of Montgomery multiplier requires less hardware, less area and it is faster, more cost effective than the existing one. It has been found that the proposed reversible sequential designs are far better than the existing ones in terms of number of gates needed, number of garbage outputs produced and quantum cost required. A cryptosystem for protection of power analysis attack of DPA is an application of reversible logic with hardware cryptography.

REFERENCES

1. Kocher, P., J. Jaffe and B. Jun, 1999. Differential power analysis. Proc. Advances in Cryptology-CRYPTO '99, Lecture Notes Comput. Sci., 1666: 388-397, Springer Verlag. http://www.cryptography.com/resources/whitepapers/DPA.pdf

2. Keyes, R. and R. Landauer, 1970. Minimal energy dissipation in logic. IBM J. Res. Develop., 14: 152-157.http://www.research.ibm.com/journal/rd/142/ibmrd1402H.pdf
3. Landauer, R., 1961. Irreversibility and heat generation in the computing process. IBM J. Res. Develop., 5: 183-191. http://www.research.ibm.com/journal/rd/053/ibmrd0503C.pdf
4. Bennett, C.H., 1973. Logical reversibility of computation. IBM J. Res. Develop., 17: 525-532. http://www.research.ibm.com/journal/rd/176/ibmrd1706G.pdf
5. Perkowski, M. et al., 2003. A hierarchical approach to computer-aided design of quantum circuits. Proceedings of 6th International Symposium on Representations and Methodology of Future Computing Technology, RM 2003, Trier, Germany, Mar. 10-11, pp. 201-209. http://web.cecs.pdx.edu/~mperkows/temp/June2/EvolvingQuantum-circuits.pdf
6. Nedjah, N. and L.M. Mourelle, 2003. Fast reconfigurable systolic hardware for modular multiplication and exponentiation. J. Syst. Architect., 49: 387-396, doi: 10.1016/S1383-7621(03)00089-4.
7. Zhang, Y.Y., Z. Li, L. Yang and S.W. Zhang, 2007. An efficient CSA architecture for montgomery modular multiplication. Microprocess. Microsyst., 31(7): 456-459, doi: 10.1016/j.micpro.2006.12.003.
8. Montgomery, P.L., 1985. Modular multiplication without trial division. Math. Comput., 44(170): 519-521. http://www.jstor.org/pss/2007970
9. Babu, H.M.H., M.R. Islam, S.M.A. Chowdhury and A.R. Chowdhury, 2004. Synthesis of full-adder circuit using reversible logic. Proceedings of 17th International Conference on VLSI Design, Jan. 5-9, Mumbai, India, pp: 757-760, IEEE Computer Society, doi: 10.1109/ICVD.2004.1261020.
10. Fredkin, E. and T. Toffoli, 1982. Conservative logic. Int. J. Theor. Phys., 21: 219-253, doi: 10.1007/BF01857727.
11. Peres, A., 1985. Reversible logic and quantum computers. Phys. Rev. A., 32: 3266-3276, doi: 10.1103/PhysRevA.32.3266.
12. Thapliyal, H. and M.B. Srinivas, 2005. A novel reversible TSG gate and its application for designing reversible carry look-ahead and other adder architectures. Proceedings of the 10th Asia-Pacific Computer Systems Architecture Conference (ACSAC 05), Lecture Notes Comput. Sci., 3740: 805-817, Springer-Verlag, doi: 10.1007/11572961_66.
13. Biswas, A.K., M.M. Hasan, A.R. Chowdhury and H.M.H. Babu, 2008. Efficient approaches for designing reversible Binary Coded Decimal adders. Microelectronics Journal, 39(12):1693-1703, doi: 10.1016/j.mejo.2008.04.003.
14. Hung, W.N.N., X. Song, G. Yang, J. Yang and M. Perkowski, 2006. Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis. IEEE Trans. Comput. Aided Des. Integrat. Circ. Syst., 25: 1652-1663, doi: 10.1109/TCAD.2005.858352.
15. Smoline, J. and D.P. DiVincenzo, 1996. Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate. Phys. Rev. A, 53(4): 2855-2856, doi: 10.1103/PhysRevA.53.2855.
16. Thapliyal, H. and M. Zwolinski, 2006. Reversible logic to cryptographic hardware: a new paradigm. Proceedings of the 49th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS '06), Aug. 6-9, Puerto Rico, 1: 342-346, doi: 10.1109/MWSCAS.2006.382067.
17. Chuang, M.L. and C.Y. Wang, 2008. Synthesis of reversible sequential elements. ACM J. Emerg. Technol. Comput. Syst., 3(4): 1-19, doi: 10.1145/1324177.1324181.
18. Yang, C.C., T.S. Chang and C.W. Jen, 1998. A new RSA cryptosystem hardware design based on Montgomery's algorithm. IEEE Trans. Circ. Syst. II: Analog Digit. Signal Process., 45: 908-913. http://ieeexplore.ieee.org/iel4/82/15144/x0098394.pdf
19. Feynman, R., 1985. Quantum mechanical computers. Optics News, 11: 11-20. Also in Foundations of Physics, 16(6): 507-531, 1986, doi: 10.1007/BF01886518.
20. Haghparast, M. and Keivan Navi, 2008. A novel reversible BCD adder for nanotechnology based systems. Am. J. Applied Sci., 5: 282-288. http://www.scipub.org/fulltext/ajas/ajas53282-288.pdf