International Journal of the Physical Sciences Vol. 6(24), pp. 5671-5681, 16 October, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.981
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

A novel nanometric fault tolerant reversible divider

Faraz Dastan[1] and Majid Haghparast[2]*

[1]Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran.
[2]Department of Computer Engineering, Shahre-Rey Branch, Islamic Azad University, Tehran, Iran

Accepted 15 August, 2011

Quantum and reversible logic circuits have more advantages than the common circuits, like low power consumption. These circuits are good choice to design future computers. One of the important issues in reversible logic is parity preservation. If parity of inputs and outputs are equal in reversible gate, this gate will be parity preserve. Reversible circuits made by these gates are parity preserve. In this paper we propose a new fault tolerant reversible divider. The proposed fault tolerant reversible divider is the first effort to design fault tolerant reversible division circuit. In this circuit, we use some fault tolerant reversible components like fault tolerant reversible parallel adder, fault tolerant reversible shift register and fault tolerant reversible n-bit register. Hence, we also propose a new fault tolerant reversible full adder, a new fault tolerant reversible n+1-bit parallel adder and a new basic cell for PIPO fault tolerant reversible left-shift register. These parity preserving reversible components are also proposed for the first time in the literature. All the scales are in the nanometric area.

Key words: Reversible logic**,** parity preservation, fault tolerant, nanometric circuits, reversible divider.

INTRODUCTION

Energy consumption is an important factor in designing quantum and reversible logic circuits. Many researches are done in the field of VLSI design (Arabshahi, 2011; Singh and Sharma, 2011; Wu et al., 2011; Chang et al., 2011). Ordinary irreversible logic circuits have energy dissipation due to information loss. Based on Landauer (1961) research, amount of energy that is wasted for every irreversible bit operation is at least "KTln2" joules, where "K" is a Boltzmann's constant (K = 1.3806505 × 10^{-23} m 2^{-2K-1} kg) and T is temperature at which operation is performed (Landauer, 1961; Parhami, 2006). For problems like this, scientists have proposed a new logic that is called reversible logic. Reversible logic circuits are developed in such a way that inputs can be obtained from outputs. In this case, KTLn2 energy is not dissipated (Bennett, 1973; Hayes, 2006). In reversible circuits, number of inputs and number of outputs are equal, we have computation history and we can return back to the every computation point easily (Ashkar and Haghparast, 2011). In addition to these, reversible logic is a good choice for optical computing, nanotechnology based systems and quantum computing. Quantum computing without using reversible logic is not possible (Haghparast and Patiar, 2011). Reversible circuits have some properties, for example in these circuits, fan out is not allowed (Parhami, 2006; Perkowski et al., 2001; Haghparast and Navi, 2008b). Reversible circuits are composed of gates, called reversible gates (Vasudevan et al., 2004). When parity of inputs and outputs are equal in reversible gate, this gate is parity preserve. Reversible gate that has parity preserving property is a fault tolerant gate. Reversible circuits which are composed of these fault tolerant gates are fault tolerant reversible circuits. Computer processor is formed from a large number of units to execution the instructions. Arithmetic units are important units in computer hardware because they have many uses in computer systems. Therefore, designing reversible logic circuits for arithmetic units is very important, nowadays. One of these arithmetic units is division unit. Division Circuit is slightly complicated in the computer hardware. There is one proposed reversible division circuit in previous papers (Nayeem et al., 2009) but it is not fault tolerant. In this paper, we propose a new fault tolerant reversible division circuit. The proposed reversible division circuit is the first effort to design fault tolerant reversible division circuit. The proposed reversible

*Corresponding author. E-mail: haghparast@iausr.ac.ir.

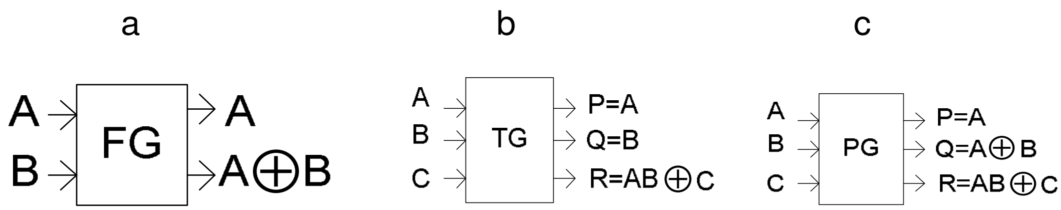

Figure 1. Some useful reversible logic gates: (a) Feynman gate, (b) Toffoli gate and (c) Peres gate.

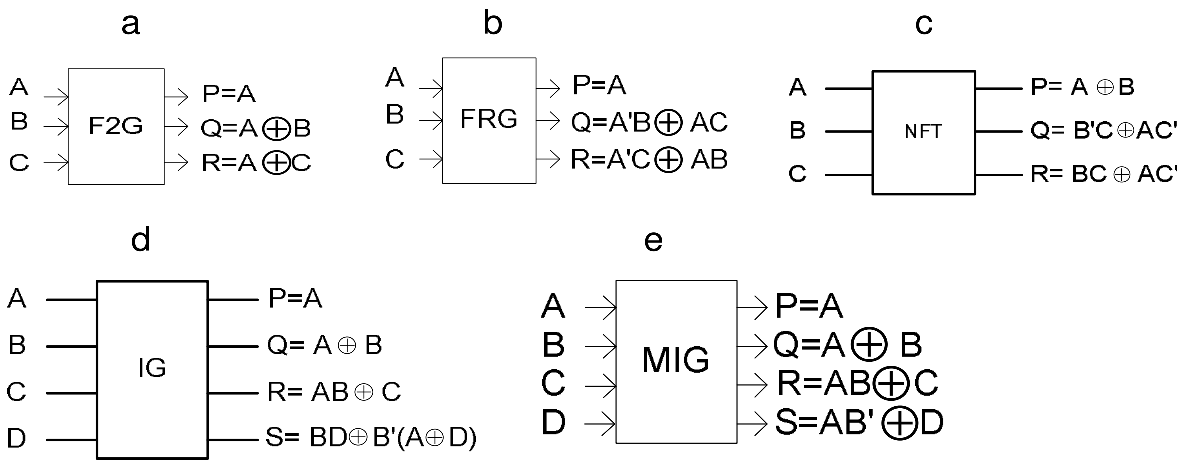

Figure 2. Parity preserving reversible logic gates: (a) Feynman double gate, (b) Fredkin gate (Fredkin and Toffoli, 1982), (c) NFT gate (d) Islam gate and (e) modified Islam gate.

fault tolerant divider composed of fault tolerant reversible components like fault tolerant reversible multiplexer, fault tolerant reversible parallel in parallel out (PIPO) left-shift register, fault tolerant reversible register and fault tolerant reversible parallel adder. Thus, in this paper we propose a new fault tolerant reversible full adder, a new fault tolerant reversible n+1-bit parallel adder and a new basic cell for fault tolerant reversible PIPO left-shift register.

Background

Definitions

Reversible logic: an "n" input and "n" output function 'F', is reversible if there is one to one correspondence between inputs and outputs. Therefore we can determine the input vector from output vector uniquely (Haghparast et al., 2008).

Garbage output: If the output of gate is not used for further computation, this output is called garbage output.

Constant inputs: Constant inputs are the inputs that are either set to zero or one (Haghparast et al., 2009).

Quantum cost: Number of 1 × 1 or 2 × 2 reversible logic gates which are needed to make the reversible circuit is called quantum cost (QC).

Parity preserving reversible gate: If parity of inputs and parity of outputs are equal in reversible gate, this gate is called parity preserve. Reversible gate is "fault tolerant" if it is parity preserve.

Reversible logic gates

A reversible 'n × n' gate can be represented as follow:

$I_v= (I_1, I_2, I_3\ldots, I_n)$ $O_v= (O_1, O2, O3\ldots, On)$

"Iv" is the input vector and "Ov" is the output vector of reversible gate. In previous papers, several reversible gates are introduced. From among them, Feynman gate (FG) (Feynman, 1985) (Figure 1a), Toffoli gate (TG) (Figure 1b) (Toffoli, 1980) and Peres gate (PG) (Peres, 1985) (Figure 1c), are some useful gates.

Parity preserving reversible logic gates

Parity checking is an important method for error detection in digital logic systems (Haghparast and Navi, 2008; Haghparast and Navi, 2008a). In Figure 2 some reversible logic gates are shown. All of these reversible logic

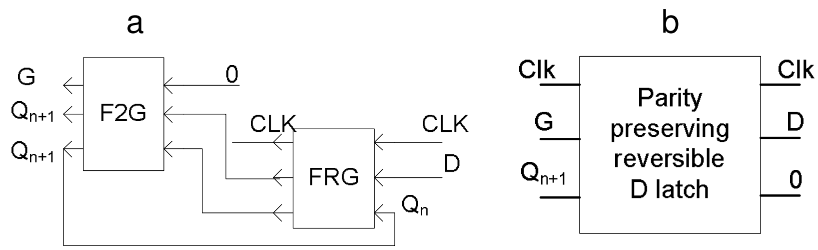

Figure 3. Parity preserving reversible D latch (Haghparast and Navi, 2011): (a) schematic and (b) block diagram.

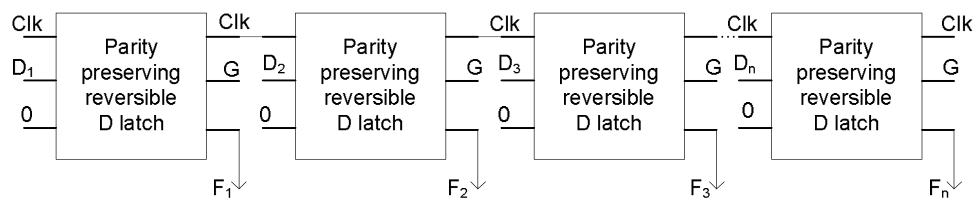

Figure 4. Parity preserving reversible register.

logic gates are parity preserve and consequently are fault tolerant reversible logic gates.

The Feynman double gate (F2G) gate is a Feynman gate with an extra input and one more output. Like these, the Islam gate (IG) (Islam and Begum, 2008; Islam et al., 2009), modified Islam gate (MIG) gates (Islam et al., 2009b) and NFT (Haghparast and Navi, 2008) are parity preserving reversible gates. Quantum cost of F2G is 2, QC of FRG is 5, QC of NFT is 5 and QC of IG is 7 (Haghparast and Navi, 2011).

Principles of division

In division process there are the parameters, dividend (K), divisor (L), remainder (M) and quotient (N). Dividend and divisor are inputs of the division process; quotient and remainder are outputs of the division process. Clearly we have $K = N \times L + M$ (Nayeem et al., 2009).

For division, we use shift/subtract division algorithm. In this algorithm, like multiplication that can be done by repeated additions, the division is done by repeated subtractions (Parhami, 2000). In every step j, we can shift the divisor j bits to the right; by this operation the divisor will be $2^{-j}L$ and then the subtract operation can be done. If partial remainder M_j bigger than $2^{-j}L$, in this case the bit n_j of the quotient is set to 1, otherwise is set to 0. Partial remainder of the next step is calculated as follows:

$$M_{j+1} = M_j - n_j \times 2^{-j} \times L \quad (1)$$

Instead of shifting right the divisor, we can shift left the partial remainder and then subtract the divisor from it; therefore like Equation 1, $M_{j+1} = 2M_j - n_j \times L$ can be inference (Nayeem et al., 2009; Hayes, 1998).

Division operation can be done by two approaches: restoring or non-restoring. In restoring division, when in every step $M_{j+1} = 2M_j - L$ was done and result was negative, the partial remainder is restored by performing this operation:

$M_{j+1} = M_{j+1}+L$ (Nayeem et al., 2009).

In non-restoring method, if subtraction result is negative, the restoring operation can be done after the shift left operation; thus we have $M_{j+1} = 2M_j+L$. By this, we can combine the operations $M_j = M_j+L$ and $M_{j+1}=2M_j-L$ (Nayeem et al., 2009). In this case, when next quotient bit is 1, the next partial remainder is calculated by subtraction operation, otherwise if next quotient bit is 0, the addition of divisor to the partial remainder is done for calculating next step. In the case that the last quotient bit is 0, the final remainder must be correct by adding the divisor to the partial remainder (Nayeem et al., 2009; Hayes, 1998).

Required components for fault tolerant reversible divider

Parity peserving reversible register

Parity peserving reversible D latch (Figure 3) (Haghparast and Navi, 2011) is used to implement the parity peserving reversible register. Parity peserving reversible register is shown in Figure 4.

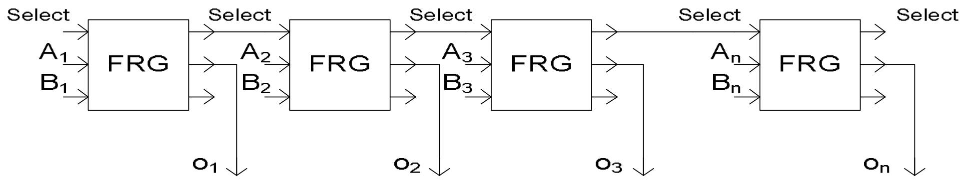

Figure 5. Fault tolerant reversible multiplexer (Nayeem et al., 2009).

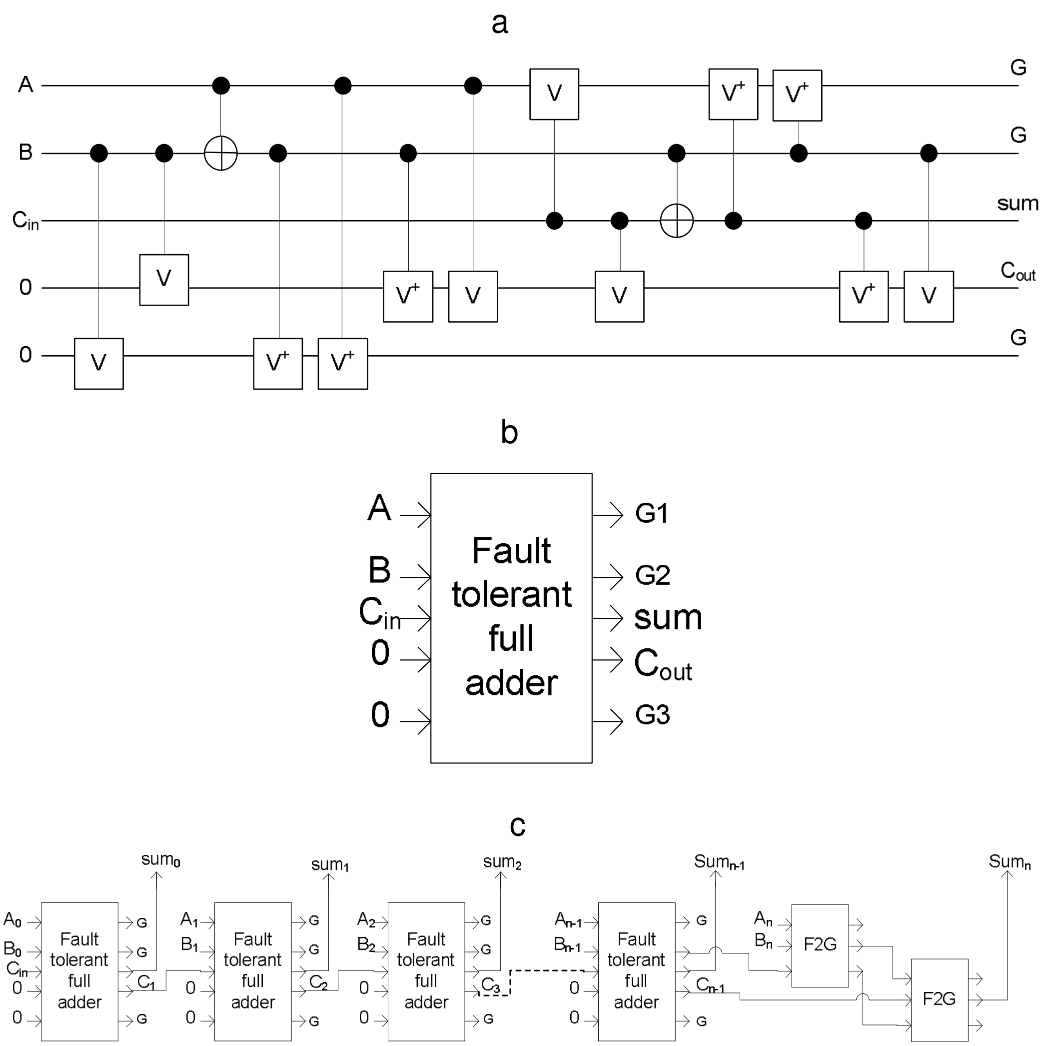

Figure 6. The proposed fault tolerant reversible circuits: (a) proposed fault tolerant reversible full adder, (b) block diagram of the proposed fault tolerant reversible full adder and (c) the proposed n+1-bit fault tolerant reversible parallel adder.

Fault tolerant reversible multiplexer (MUX)

The Figure 5 illustrates a two-input MUX with one select line. The FRG gates are used to realize this MUX. The Lines A (A_1, A2,…, An) and B (B_1, B2…, B3) are MUX inputs. If select line is 0 then O = A (MUX output is A) and if select line is 1, then O = B (MUX output is B) (Nayeem et al., 2009).

Fault tolerant reversible n+1-bit parallel adder

The proposed fault tolerant reversible full adder is shown in Figure 6a. In Figure 6b block diagram of the proposed

Table 1. Control inputs of reversible PIPO left-shift register (Nayeem et al., 2009).

SV	E	Final Output Q_i^+
0	0	Q_{i-1} (left shift)
0	1	I_i (parallel load)
1	×	Q_i (no change)

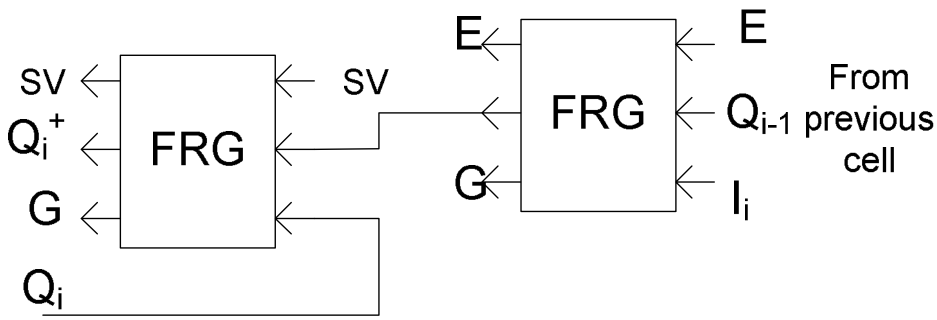

Figure 7. Implementation of Equation 2 with reversible logic gates (Nayeem et al., 2009).

fault tolerant reversible full adder is shown. Figure 6c shows the proposed n+1-bit fault tolerant reversible parallel adder. In this unit for last bit position of parallel adder that requires computing two XOR (Nayeem et al., 2009), 2 F2G gate is used and C_{out} of this parallel adder is ignored.

Fault tolerant reversible PIPO left-shift register

Parallel data bits can be loaded into the reversible PIPO left-shift register. This data bits also can be shifted left and appear in outputs. Control inputs of reversible left-shift register are shown in Table 1. These inputs are used to select the operation of the reversible PIPO left-shift register.

According to Table 1, during clock pulse, when SV and E are 0 (low), left-shift is performed. When SV is 0 and E is 1 (high), parallel load is performed. Therefore, input bits I_i are loaded into the left-shift register and outputs are available from the Q output. When SV is 1, the reversible PIPO left-shift register saves its current value; in other words, the reversible PIPO left-shift register do not perform anything (inactive) when SV is 1. Q_i^+ can be obtained as follow (Nayeem et al., 2009):

$$Q_i^+ = SV'.E.\ I_i + SV'.E'.\ Q_{i-1} + SV.\ Q_i \qquad (2)$$

Figure 7 shows the implementation of Equation 2. Reversible circuit that is used to implement the Equation 2 in figure 7 is a fault tolerant reversible circuit, because it is composed of FRG gates.

In Figure 8a, the proposed basic cell of fault tolerant reversible PIPO left-shift register is shown. Figure 8b shows a block diagram of the proposed fault tolerant reversible PIPO left-shift register basic cell. Figure 9 shows an n-bit fault tolerant reversible PIPO left-shift register.

PROPOSED FAULT TOLERANT REVERSIBLE DIVIDER

Approach 1

Figure 11 shows the proposed fault tolerant reversible divider. This circuit includes two fault tolerant reversible PIPO left-shift register (n-bit and n+2-bit), two fault tolerant reversible MUX (n-bit and n+1-bit), one n-bit fault tolerant reversible register for holding the divisor and one n+1-bit fault tolerant reversible parallel adder for performing addition or subtraction. Like division process in (Nayeem et al., 2009), initially:

$A(A_{n-1},A_{n-2}\ldots,A_1)$ = high order half of the dividend,

$S = 0$, $D(D_{n-1},D_{n-2}\ldots,D_1)$ = low order half of the dividend,

$V(V_{n-1},V_{n-2}\ldots,V_1)$ = divisor (dividend has 2n bits and divisor has n bits).

CTR is also a control unit signal that its value is 0. The select line of the two MUX's is 1 and thus, n-bit MUX, selects low order half of the dividend (D) and n+1-bit MUX, selects S = 0 and A = high order half of the dividend. During clock, when E = 1 and SV2 = 0, the output of n+1-bit MUX and $S_1 = 1$ is loaded into the n+2-

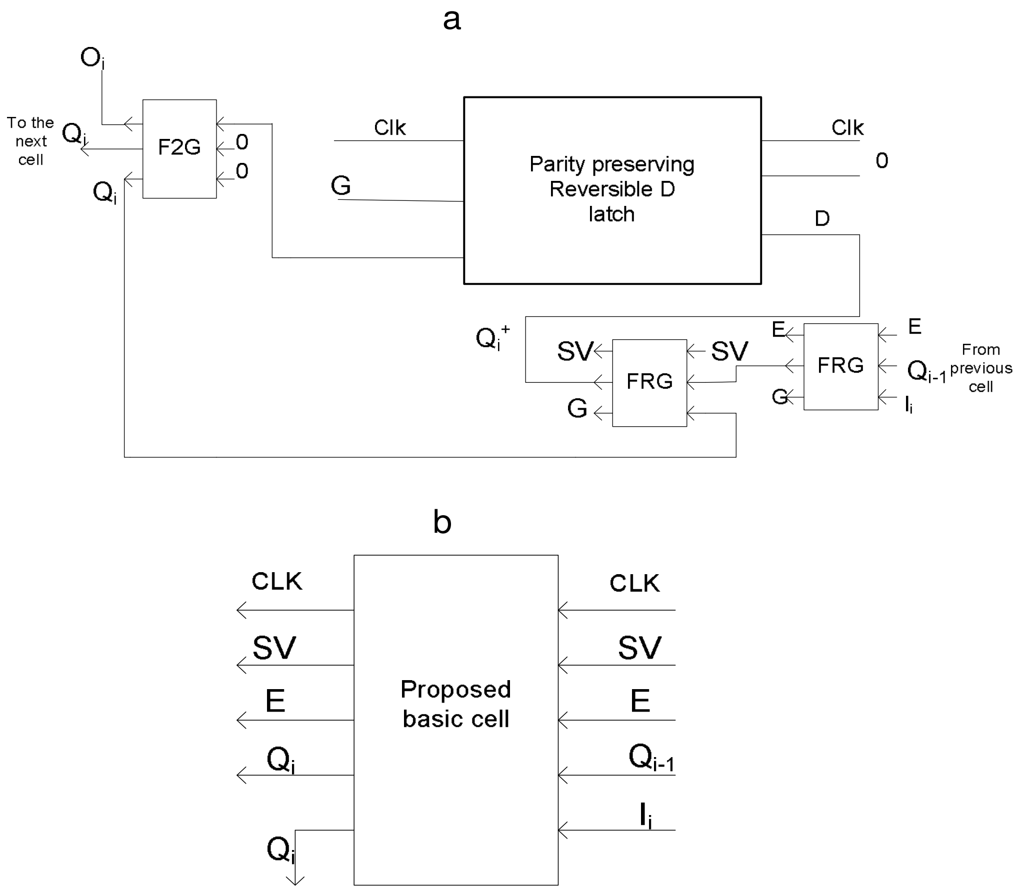

Figure 8. Proposed basic cell for fault tolerant reversible PIPO left-shift register: (a) schematic and (b) block diagram.

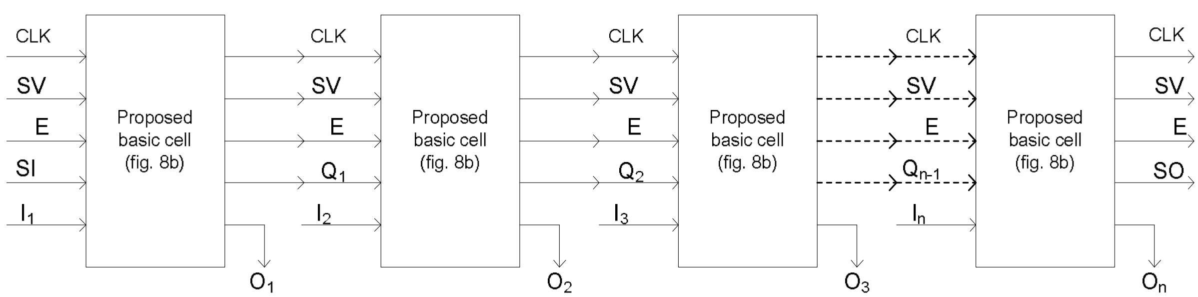

Figure 9. An n-bit fault tolerant reversible PIPO left-shift register.

bit fault tolerant reversible PIPO left-shift register (S_1 S A) and when SV1 = 0, outputs of n-bit MUX is loaded into the n-bit fault tolerant reversible PIPO left-shift register (Q).

Division process realized as follow: after loading the initial value of dividend in to the left-shift registers and the divisor is loaded into the n-bit register, E is changed to 0 and select line of the MUX's is chanced to 0; therefore left-shift operation can be done (one bit) in the next clock. Addition and subtraction in the parallel adder is performed on the S.A and V. In other words, for addition S.A + V and for subtraction S.A - V is performed. Add operation is needed to restore the partial remainder. To determine the bits of the quotient, the complement of the most significant bit (MSB) is used. This is realized as follows: during clock, the complement of MSB is loaded in

Dividend= 10001101 Divisor= 1001

	S1	S A3 A2 A1 A0	Q3 Q2 Q1 Q0
initialy	1	01000	1101
Shift-left	0	01000	101-
Subtract		10111	
Insert ‘1’ to Q_0	×	01000	1011
Shift-left	0	10001	011-
Subtract		10111	
Insert ‘1’ to Q_0	×	01000	0111
Shift-left	0	10000	111-
Subtract		10111	
Insert ‘1’ to Q_0	×	00111	1111
Shift-left	0	01111	111-
Subtract		10111	
Insert ‘1’ to Q_0		00110 = Final remainder	1111 = Qutient

Final remainder restoration not necessery

Figure 10. Numerical example of division process.

to Q0 bit position of the n-bit fault tolerant reversible PIPO left-shift register to form next bit of the quotient. Simultaneously, output (n+1-bit) of the parallel adder is loaded in to S.A (when select = 0 and E = 1).

To maintain the quotient bits into the Q, we need 2n+1 clock pulses. After 2n+1 clock pulse, CTR is changed 1. This leads to SV1 = 1 and thus, the Q register stores the final quotient. After 2n+1 clock pulse, in a situation that SO of the n-bit left-shift register enters to the LSB bit of the n+2-bit left-shift register and value of the S enters to the S_1. S_1 is used to select the parallel adder operation. It means that when S_1 is 1, the parallel adder performs add operation and when S_1 is 0 the parallel adder performs subtract operation. Addition and subtraction in the parallel“(S = 0 and CTR = 1) $\rightarrow$ SV2 = 1” (because FRG gate performs S'.CTR), therefore ‘A’ holds the final remainder. But if S = 1, the remainder must be restored. Restoration performed by add operation in the parallel adder (S.A+V) and therefore, at the next clock, correct value of the remainder is loaded into the S.A. after restoration, S = 0 and consequently SV2 = 1 and final remainder is stored in ‘A’. Figure 10 shows a numerical example of division process.

Approach 2

In approach 2, division process is like approach 1, but architecture of the division circuit is slightly changed. Instead of n+2-bit fault tolerant reversible PIPO left-shift register, we use n+1-bit fault tolerant reversible PIPO left-shift register and one D latch to maintain S_1. One FRG

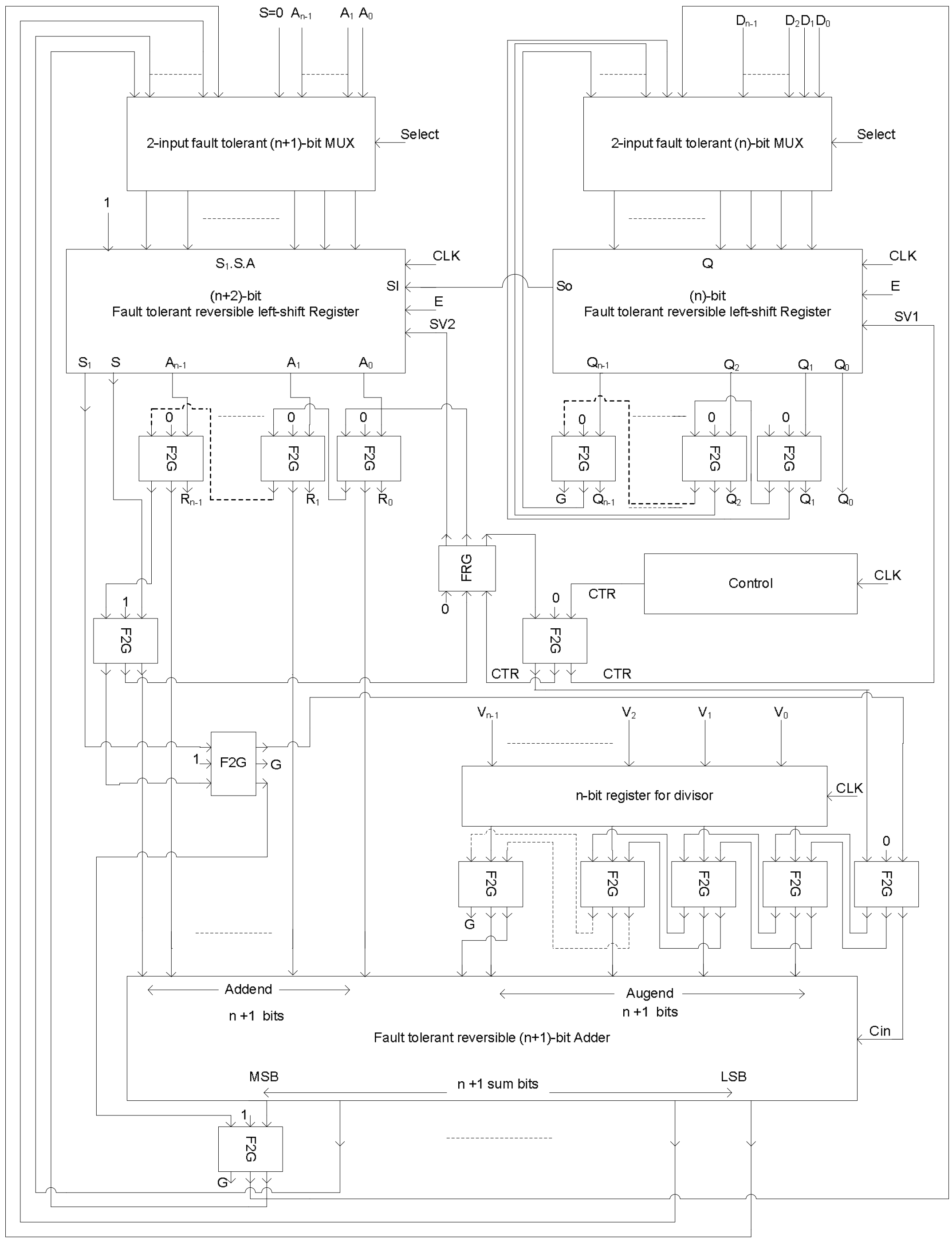

Figure 11. Proposed fault tolerant reversible divider (Approach 1).

gate is used as a one-bit, two input MUX.

Figure 12 shows the proposed fault tolerant reversible divider 2. According to the Figure 12, SO of the n+1-bit fault tolerant reversible PIPO left-shift register is connected to the input of D latch. Output of the D latch (S_1) becomes complement and determines the parallel adder operation. In this approach, after 2n+1 clock pulse division process reaches to the end, control signal generator, changes the CTR to 1; in this situation if S = 0 then SV2 will be 1 and thus, the partial remainder can save in the S.A left-shift register. But if S = 1 (after 2n+1 clock pulse), the partial remainder must be restored by adding the partial remainder to the divisor. CTR, also, is connected to the select line of the one bit MUX (FRG

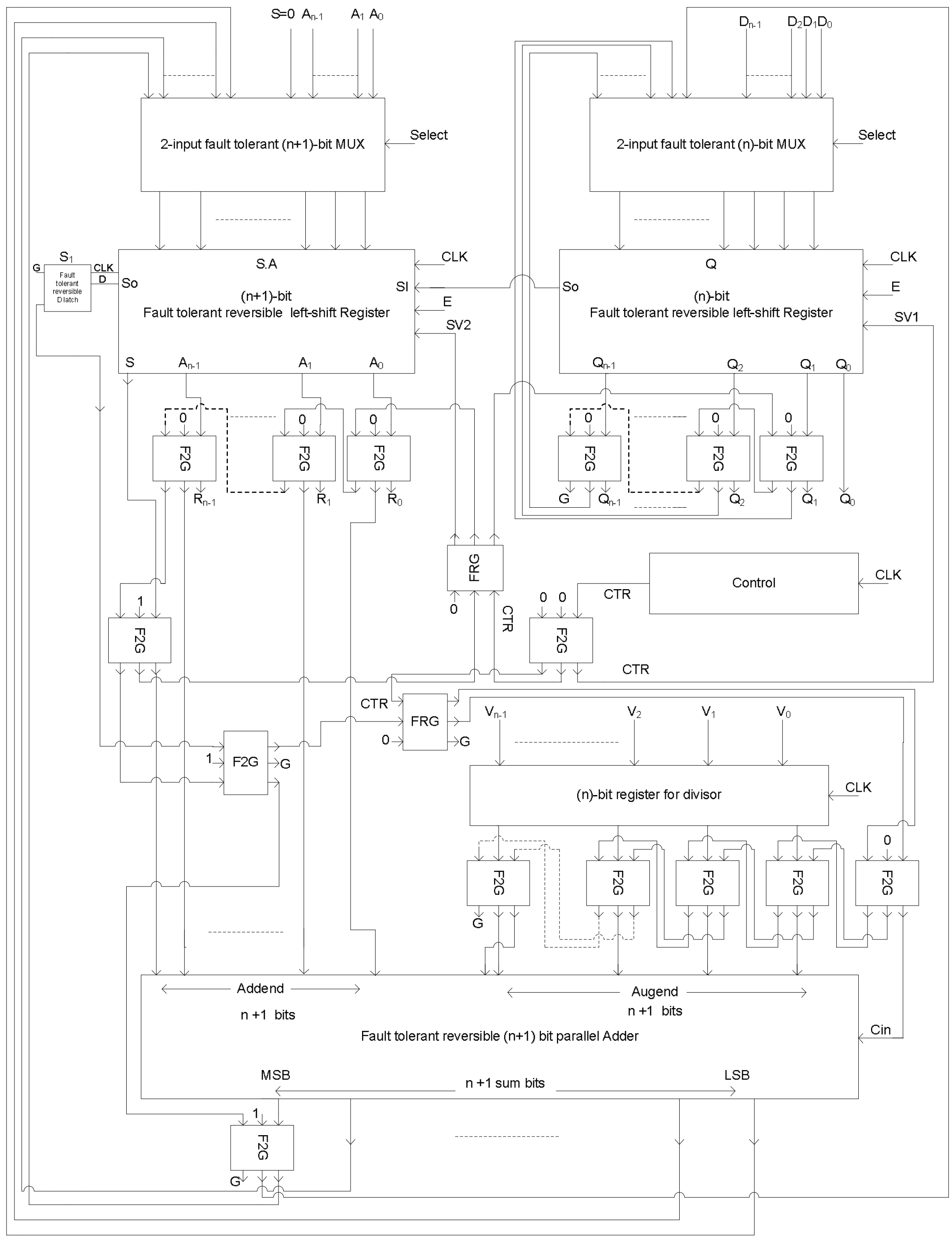

Figure 12. Proposed fault tolerant reversible divider (Approach 2).

gate) and when CTR is changed to 1, MUX inserts 0 to its output and thus, parallel adder operates a add operation and at the next clock correct value of the remainder is stored in ‘A’. By approach 2, total number of reversible gates, garbage outputs, quantum cost and number of constant inputs in fault tolerant reversible divider are decreased.

EVALUATION OF THE PROPOSED FAULT TOLERANT REVERSIBLE DIVIDER

For approach 1 of the proposed fault tolerant reversible divider we have:

1. n+1-bit fault tolerant reversible MUX: number of garbage

Table 2. Characteristics of the proposed n-bit fault tolerant reversible divider.

	Garbage outputs	Quantum cost	Constant inputs
Approach 1	12n+18	75n+60	11n+14
Approach 2	12n+16	75n+53	11n+12

outputs = n+1, quantum cost = 5n+5, number of constant inputs = 1
2. n-bit fault tolerant reversible MUX: number of garbage outputs = n, quantum cost = 5n
3. n+2-bit fault tolerant reversible PIPO left-shift register: number of garbage outputs = 3n+9, quantum cost = 19n+38, number of constant inputs= 3n+7
4. n-bit fault tolerant reversible PIPO left-shift register: number of garbage outputs = 3n+2, quantum cost = 19n, number of constant inputs = 3n
5. n-bit fault tolerant reversible register: number of garbage outputs = n, quantum cost = 7n, number of constant inputs = n
6. n+1-bit fault tolerant reversible parallel adder: number of garbage outputs = 3n+2, quantum cost = 14n+4, number of constant inputs = 2n
7. Other gates: (number of F2G gate = 3n+4 and number of FRG gate = 1) number of garbage outputs = 4, quantum cost = 6n+13, number of constant inputs = 2n+6.

For approach 2 of the proposed fault tolerant reversible divider we have:

1. n+1-bit fault tolerant reversible MUX: number of garbage outputs = n+1, quantum cost = 5n+5, number of constant inputs = 1
2. n-bit fault tolerant reversible MUX: number of garbage outputs = n, quantum cost = 5n
3. n+1-bit fault tolerant reversible PIPO left-shift register: number of garbage outputs = 3n+5, quantum cost= 19n+19, number of constant inputs = 3n+3
4. n-bit fault tolerant reversible PIPO left-shift register: number of garbage outputs = 3n+2, quantum cost = 19n, number of constant inputs = 3n
5. n-bit fault tolerant reversible register: number of garbage outputs = n, quantum cost = 7n, number of constant inputs = n
6. n+1-bit fault tolerant reversible parallel adder: number of garbage outputs = 3n+2, quantum cost = 14n+4, number of constant inputs = 2n
7. Other gates: (number of F2G gate = 3n+4, number of FRG gate = 2 and one D latch) number of garbage outputs = 6, quantum cost = 6n+25, number of constant inputs = 2n+8

Table 2 shows three characteristics of the proposed (approach 1 and 2) n-bit fault tolerant reversible divider.

Conclusion

Arithmetic units in computer systems have much usage; hence, designing these units is very important. Computer designers are interested to design reversible logic circuits with reversible gates and designing fault tolerant reversible circuit is important nowadays. In this paper we proposed two approaches for designing fault tolerant reversible divider. This fault tolerant reversible divider is the first fault tolerant reversible divider that has been proposed so far. Table 2 shows characteristics of the proposed fault tolerant reversible divider in two approaches. Also in this paper a new fault tolerant reversible full adder, a new fault tolerant reversible n+1-bit parallel adder and a new basic cell for fault tolerant reversible PIPO left-shift register are proposed. These units are used in the proposed fault tolerant reversible divider. All the scales are in the nanometric area. The proposed fault tolerant reversible divider with the other fault tolerant reversible circuits are used to generate large fault tolerant reversible systems in nanotechnology. By the use of these systems, the fault tolerant quantum computers can be realized. In the future works we will concentrate on the design of signed dividers.

Nomenclatures: **CTR,** Control; **⊕,** exclusive or; **F2G,** Feynman double gate; **FG,** Feynman gate; **FRG,** Fredkin gate; **IG,** Islam gate; **LSB,** least significant bit; **MIG,** modified Islam gate; **MUX,** multiplexer; **NFT,** new fault tolerant gate; **PG,** Peres gate; **PIPO,** parallel in parallel out; **QC,** quantum cost; **TG,** Toffoli gate.

REFERENCES

Arabshahi H (2011). Comparison of low field electron transport characteristics in Ge and Si semiconductors and effects of neutron energy deposition on their crystal structure. Int. J. Phys. Sci., 6(9): 2327-2334.
Ashkar M, Haghparast M (2011). A Novel Design of Nanometric Reversible Cache Memory. Sci. Res. Essays, 6(14): 3034-3040.
Bennett CH (1973). Logical reversibility of computation. IBM J. Res. Dev., 17: 525-532.
Chang WC, Su SC, Ma KH (2011). An efficient inverter circuit design for driving the ultrasonic welding transducer. Int. J. Phys. Sci., 6(6): 1332-1341.
Feynman R (1985). Quantum mechanical computers. Optics News, 11: 11-20.
Fredkin E, Toffoli T (1982). Conservative logic. Int. J. Theor. Phys., 21: 219-53.
Haghparast M, Jassbi SJ, Navi K, Hashemipour O (2008). Design of a novel reversible multiplier circuit using HNG gate in nanotechnology.

World Appl. Sci. J., 3(6): 974-978.

Haghparast M, Mohammadi M, Navi K, Eshghi M (2009). Optimized reversible multiplier circuit. J. Circuits Syst. Comp., 18(2): 311-323.

Haghparast M, Navi K (2008). A Novel Fault Tolerant Reversible Gate for Nanotechnology Based Systems. Am. J. Appl. Sci., 5(5): 519-523.

Haghparast M, Navi K (2008a). Design of a Novel Fault Tolerant Reversible Full Adder for Nanotechnology Based Systems. World Appl. Sci. J., 3(1): 114-118.

Haghparast M, Navi K (2008b). A novel reversible BCD adder for nanotechnology based systems. Am. J. Appl., Sci., 5(3): 282-288.

Haghparast M, Navi K (2011). Novel Reversible Fault Tolerant Error Coding and Detection Circuits. Int. J. Quantum Inf., 9(2): 723-738.

Haghparast M, Patiar A (2011). Novel nanometric reversible saturating adder. Int. J. Phys. Sci., 6(9): 2321-2326.

Hayes B (2006). Reversible engineering. Am. Sci., 94: 107-111.

Hayes JP (1998). Computer Architecture and Organization. 3rd Edition, McGraw-Hill, New York.

Islam MS, Begum Z (2008). Reversible logic synthesis of fault tolerant carry skip BCD adder. Bangladesh Acad. Sci. J., 32(2): 193-200.

Islam MS, Rahman MM, Begum Z, Hafiz MZ (2009). Fault Tolerant Reversible Logic Synthesis: Carry Look-Ahead and Carry-Skip Adders. IEEE Int. Conf. Adv. Comput. Tools Eng. Appl., 396-401.

Islam MS, Rahman MM, Begum Z, Hafiz MZ, Mahmud AA (2009b). Synthesis of fault tolerant reversible logic circuits. Proc. IEEE International Conference on Testing and Diagnosis, Chengdu, China.

Landauer R (1961). Irreversibility and heat generation in the computing process. IBM J. Res. Dev., 5: 183-191.

Nayeem NM, Hossain MA, Haque MM, Jamal L, Babu HMH (2009). Novel Reversible Division Hardware. IEEE Int. Midwest Symp. Circuits Syst., 1134-1138.

Peres A (1985). Reversible logic and quantum computers. Phy. Rev., 32: 3266-3276.

Parhami B (2000). Computer Arithmetic: Algorithms and Hardware designs. Oxford University Press, New York.

Parhami B (2006). Fault Tolerant Reversible Circuits. Proc. ACSSC, pp. 1726-1729.

Perkowski M, Al-Rabadi A, Kerntopf P, Buller A, Chrzanowska-Jeske M, Mishchenko A, Khan MA, Coppola A, Yanushkevich S, Shmerko V, Jozwiak L (2001). A general decomposition for reversible logic. Proc. RM, Starkville, 119-138.

Singh PK, Sharma S (2011). Substrate noise coupling in NMOS transistor for RF/analog circuits. Int. J. Phys. Sci., 6(9): 2285-2293.

Toffoli T (1980). Reversible computing. Tech Memo MIT/LCS/TM-151. MIT Lab for Computer Science.

Vasudevan DP, Lala PK, Parkerson JP (2004). Online testable reversible logic circuit design using NAND blocks. Proc. DFTVS, pp. 324-331.

Wu L, Guan Y, Du Y, Zhou SH, Pan W (2011). Deterioration analysis of aluminum electrolytic capacitor for DC-DC converter. Int. J. Phys. Sci., 6(7): 1653-1664.