Original scientific paper

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 48, No. 3(2018), 161 – 171

Design of Fault-Tolerant Reversible Floating Point Division

A. Kamaraj[1], P. Marichamy[2]

[1]Department of ECE, Mepco Schlenk Engineering College, Sivakasi, India
[2]Department of ECE, PSR Engineering College, Sivakasi, India

Abstract: In semiconductor industries power dissipation and the size of the computational devices are playing a major role. Size of a single transistor may limit the scaling of semiconductor devices. In turn, an alternative technology is needed for computational devices; one such technology is Reversible Logic. In this paper, a new set of reversible gates named as KMD Gates are proposed, they are capable of producing many logical functions compared to the conventionally available reversible gates. The proposed gates satisfy the reversibility and universality properties of reversible logic. In addition, these gates are having parity preservation, so they are fault-tolerant. A n-bit fault-tolerant reversible floating point division unit (FTRFPD) is designed in IEEE 754 single precision standard using the proposed fault-tolerant KMD reversible gates. This FTRFPD has parallel adder, latch, multiplexer, shift register, rounding and normalization register. All the functional blocks are fault-tolerant in nature as they are, they are constructed from the Fault-Tolerant Gates. The FTRFPD is capable of dividing two numbers using the non-restoring algorithm. Quantum Cellular Automata (QCA) is incorporated for validating the functionality of the reversible logic gates and division unit. The QCA based simulation results confirm that the designed unit is having reduction in Quantum Cost by 9.85%, in Delay by 29.63% and in Number of Gates by 33.54 % over the existing designs.

Keywords: Reversible Logic; Quantum Cost; Delay.

Načrtovanje proti napakam odpornega reverznega deljenja s plavajočo

Izvleček: Poraba moči in velikost računskih elementov igrajo pomembno vlogo v polprevodniški industriji. Velikost posameznega tranzistorja lahko omejuje velikost poprevodniške naprave, zaradi česar je potrebna drugačna tehnologija za računske elemente, kot na primer reverzibilna logika. Članek predlaga nov set reverzibilnih vrat kot KMD vrat, ki omogočajo več logičnih operacij kot klasična vrata. Vrata zagotavljajo reverzibilnost in univerzalnost reverzibilne logike. Poleg tega predlagana vrata ohranjajo polariteto, kar pomeni, da so odporna proti napakam. Enota za deljenje s plavajočo vejico (FTRFPD) je načrtana v IEEE 785 standardu z enojno natančnostjo z uporabo predlaganih KMD vrat. FTEDPD ima paralelni množilnik, zapah, miltiplekser, pomikalni regiter ter zaokroževalni in narmalizacijski register. Vsi funkcijski blogi so odporni na napake saj vrebujejo proti napakam odporna vrata. FTEDPD lahko deli dve števili brez algoritma obnovitve. Za validacijo vrat in deljenja je uporabljen QCA (Quantum Cellular Automata). QCA simulacije potrjujejo zmanjšanje stoška kvanta za 9.85%, zakasnitve za 29.63% in števila vrat za 33.54% glede na obstoječe dizajne.

Ključne besede: reverzibilna logika; strošek kvanta; zakasnitev.

** Corresponding Author's e-mail: kamarajvlsi@gmail.com*

1 Introduction

In modern VLSI technology, size and power dissipation are the major challenges to computational devices. The size reduction of a transistor scaling has its own physical limits. Also, smaller transistors dissipate more power, and it has other second-order effects, which affect the functionality of the circuits. Landauer has proved that the irreversible computing devices dissipates heat energy in the order of KTln2 Joules for the loss of a single bit of information (where K is Boltzmann's Constant = 1.3807 x 10^{-23} J, T is room temperature) [1]. However, Benett has shown that the reversible computation using reversible gates is the solution to this issue [2]. It has two approaches physical and logical reversibility. Logical reversibility is the ability of the device to retrieve its input and output from each other logically;

whereas Physical reversibility is provision to reverse the input and output in electronic or quantum circuits [3].

Parity checking is one of the common mechanism adapted to detect the error in the transmission and data logic. During the computation, if the parity of the input data is preserved, then there is no need for checking the intermediate stages. Thus, the parity-preserving functional units can be constructed with parity-preserving reversible gates [4].

The division is one of the most complicated functions of computer arithmetic. In this paper, we have proposed fault-tolerant reversible logic gates to construct division unit. n-bit fault-tolerant reversible floating point division (FTRFPD) circuit is constructed with these gates. The proposed design is vulnerable to errors, and it is better than the existing ones in terms of quantum costs, garbage outputs and constant inputs.

The entire paper is organized as follows: in Section II, the basic definitions and performance measuring parameters are discussed. Section III discusses the relevant work previously done by various researchers in the same field. Section IV & V deal with the proposed new fault-tolerant reversible gates and fault-tolerant reversible floating point division unit. Finally, in Section VI the results obtained for the proposed division unit are being discussed.

2 Basic definitions

In this section basic definition of QCA, QCA clocking and performance measures commonly used are discussed.

2.1 QCA Basics

The basic element of QCA is a cell, which represents a bit of information as in Fig.1a. Each cell has four metal islands known as quantum dots. In quantum dots, two electrons occupy the four dots in "diametrically opposite" positions to polarize the dot. Logic '0' or Logic'1' is represented by the polarization position of dots as shown in Fig.1b[5].

The polarization can be computed as,
$P= ((P_1+P_3)-(P_2+P_4))/(P_1+P_2+P_3+P_4)$,
where P_i= Charge in the i^{th} quantum dot.

2.2 QCA Clocking

QCA circuit's information flow is controlled by the Bennett Clocking scheme. It has four phases to process the

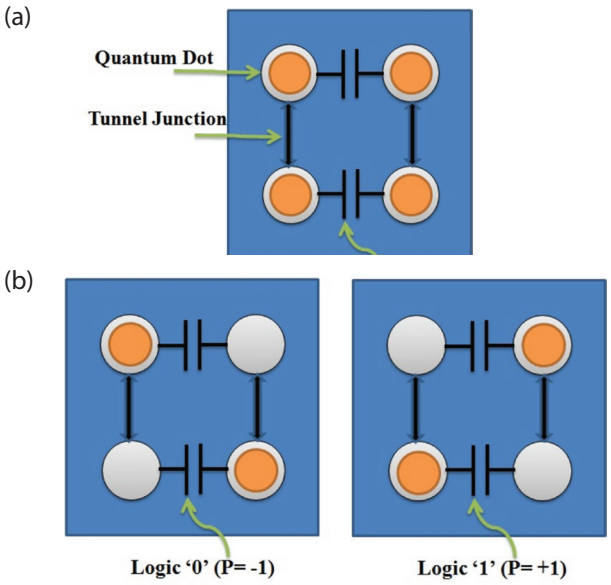


Figure 1: (a) QCA cell; (b) QCA cell polarized with logic '0' and logic '1'

information which is a switch, hold, release and relax as in Fig. 2. Each clock zone is 90-degree phase-shifted, which enables the computation to be carried out in a sequential manner [6].

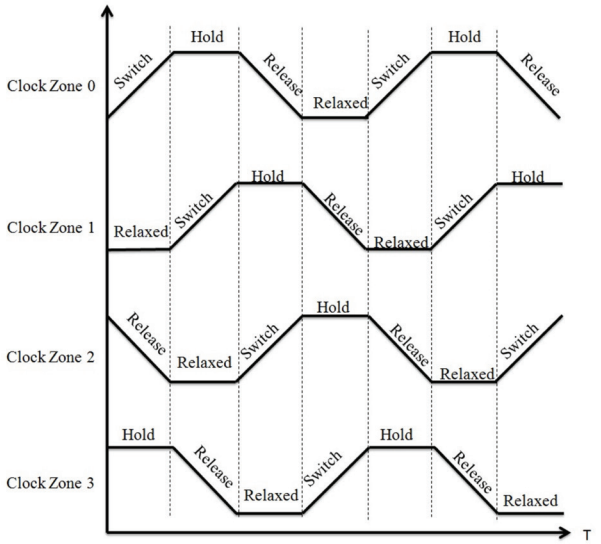


Figure 2: QCA Clocking with four phases

2.3 Performance measures

Four Performance measures encountered in this work are listed below:

- Quantum Cost: Quantum cost is calculated as the total number of 2*2 primitive gates required to derive the given reversible logic function [7]. Here the primitive gates refer to the conventional gates. They are Feynman, Toffoli, Fredkin and Peres gates.

- Garbage output: The unwanted output of the reversible circuit is known as garbage output [7].
- Constant inputs: The input set to a stable value throughout the computation of the reversible circuit is known as constant inputs [7].
- Number of gates: It is defined as the total number of reversible gates that are utilized to obtain the desired reversible logic [7].
- Delay: The delay of the reversible circuit is equal to the total number of gates in the circuit. [8]. In other words, the maximum number of gates between the input to the output of the function is the delay of the reversible circuit.
- Simple Reversible Gates: Fredkin, Feymann (CNOT), Toffoli and Peres are the most popularly used reversible gates. Toffoli is known as universal reversible 3 input gate, from this any reversible circuit can be derived [9]-[12].
- Other reversible gates: There are many other authors who proposed their own reversible gates such as DKG gate, MRG gate [13], NFT gate, TR gate and BVF gate [14]. Since, except NFT other gates are not fault-tolerant and the circuits derived from those gates are not fault-tolerant.

2.4 Fault-tolerant reversible logic

A Gate is said to be fault-tolerant reversible logic gate only when it satisfies the following three properties:

- Reversibility: In reversibility, input and output functions are uniquely derived from each other and vice versa for a reversible gate [15].
- Input ⇔Output
- Universality: When a reversible gate is able to realize NOT, NAND / AND & NOR / OR functions in the output, it satisfies universality property [15].
- Fault Tolerance: Nowadays, a reversible circuit is expected to be reliable under all environmental conditions. In other words, it should be vulnerable to the fault occurrence. A gate having the same parity as in equation (1) in its input and output is said to be a fault-tolerant gate [4].

$$I_1 \oplus I_2 \oplus I_3 \ldots\ldots \oplus I_n = O_1 \oplus O_2 \ldots\ldots \oplus O_n \quad \text{--------} \quad (1)$$

where, I_i = Inputs and O_i = Outputs of a gate

3 Related reversible logic works

A new reversible Half adder, Full Adder, Ripple carry Adders are being built using the proposed reversible logic gates. These gates satisfy the universality and reversibility conditions as a fundamental requirement for a reversible gate [16].

Fault-tolerant full adder (FTFA) is proposed using two Islam Gates (IG) which have 3 garbage outputs and 2 constant inputs. A carry skip BCD is designed with the proposed FTFA functional unit [17]. A fault-tolerant reversible adder (FTRA) is proposed in [18] and ripple carry adder (RCA), carry skip adder (CSA) and n-bit ALU are constructed using FTRA, which are fault-tolerant [18]. Low-cost parity-preserving reversible adders such as Carry look ahead adder (CLA), Carry skip adder (CSA) and BCD adder are constructed with less quantum cost and garbage output using LCG gates [19]. A fault-tolerant reversible decoder (n to 2n) is constructed using Double Fredkin and RDC gates [20].

A signed multiplier is designed with MNFT gate which is fault-tolerant. The operating speed of the multiplier is improved using Wallace tree structures [21]. A reversible single precision floating-point square root is proposed using a modified non-restoring algorithm with Reversible Controlled-Subtract-Multiplex [22].

As an initial step towards the sequential circuit design, the latches have been designed with reversible gates. RS Flip-flop is proposed in [23] with reversible gate BME and Peres. Also, D, RS and JK latches and flip-flops have been designed using MFG, FG and Toffoli reversible gates in [24].

Two approaches are proposed for n-bit fixed point division unit. Here, fault-tolerant reversible gates are utilized to make the entire division unit fault-tolerant. Both the approaches utilize nearly equal number of resources (quantum cost, garbage output etc.) [25]. A reversible floating point division is carried out with two different approaches which are conventional division and high-speed division. The results confirm that this approach is better than the previous [22] architectures in terms of quantum cost, garbage output [26]. A new fault-tolerant reversible RR gate is proposed to design a reversible division unit with fault-tolerance of n-bit. The module is constructed to meet the IEEE 754 format, which includes rounding register and normalization unit. Also, the division unit consumes less number of quantum cost, garbage output and number of gates than the previously available designs [8].

4 The proposed logic gates

A new reversible gate is introduced in two different methods: one is heroic act on the existing reversible gates and another is the creation of new gate to perform the desired operation. Fault tolerant 3*3 reversible gates are proposed to obtain the full adder function using Feynman and Fredkin gates [19]. These gates may not satisfy the universality property of a reversible gate.

Here, we propose 4 reversible gates, namely KMD gate 1, KMD Gate 2, KMD Gate 3 and KMD Gate 4 as shown in Fig. 3 (a-d). These gates satisfy the fundamental requirements (reversibility and universality) of a reversible gate [15]. In addition, they are fault-tolerant in nature, i.e. EXOR function of the inputs and the outputs are equal (parity preservation).

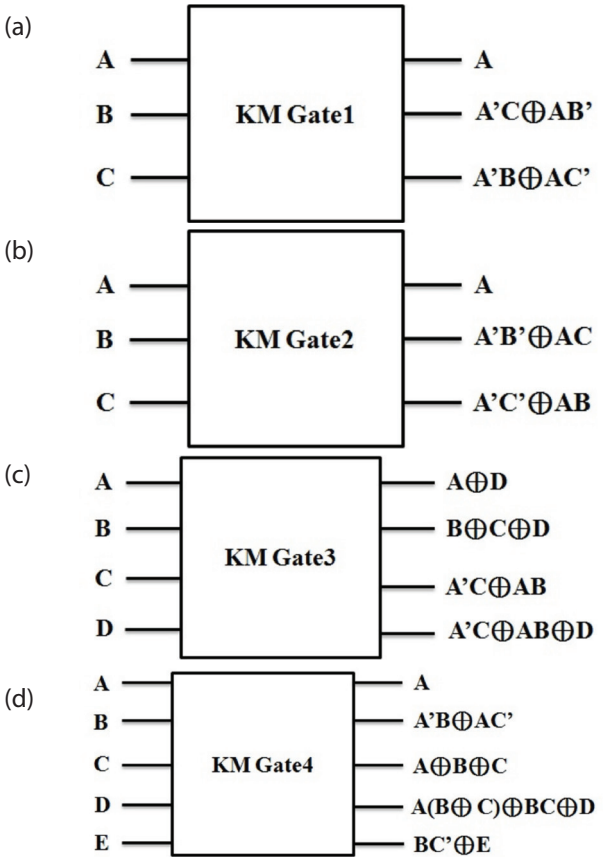


Figure 3: Block diagram (a) KMD gate 1 (b) KMD gate 2 (c) KMD gate 3 and (d) KMD gate 4

4.1 Fault-Tolerance, reversibility, and universality

The fault-tolerant (parity preservation) and reversibility characteristics of the proposed gates are satisfied for the proposed gates. The inputs and the outputs of KMD Gates are having the same priority. So, the EXOR of Inputs and Outputs gives always zero for a fault-tolerant gate.

Reversibility can be defined in two ways; one is, the computation overwriting the input vector with the output vector, and the other is, an unmodified copy of the input vector available elsewhere in the design [15].

For universality, a reversible gate must be able to produce NOT, AND & OR functions of 2-input format or it must be able to generate NOT, NAND / NOR functions of 2-input format [15]. The universality property of KMD Gates is represented in Table 1. From the above table, it is evident that all KMD Gates satisfy the universality property as stated in [15].

The proposed reversible gates can be constructed with fewer cells and occupy less area as shown in Table 2. Moreover, DKG and MRG are not fault-tolerant gates [13] [14]. But KMD Gates are fault-tolerant reversible gates. Thus, the construction of reversible circuits using these gates will have efficient fault-tolerant reversible circuits.

5 The Proposed methodology

In arithmetic operations, the basic operations are addition, subtraction, multiplication, and division. Of them, the division is the most challenging arithmetic

Table 1: Universality property of KMD gates

S. No.	Reversible Gate	Constant Input	Logic Function	Expression
	KMD Gate 1	A=1; C=0 / 1	Q=NOT (B)	Q = R′
		C=1	Q = NAND (A,B)	Q = A′ + AB′
		C=0	R = OR (A, B)	Q = A + A′B
	KMD Gate 2	B=C=0	R = NOT (A)	R = A′
		C=0	Q = NOR (A,B)	Q = A′B′ = (A+B)′
		C=1	R = AND (A,B)	R = AB
	KMD Gate 3	B=D=0; C=1	S = NOT (A)	S = A′
		B=1	R = OR (A, C)	R = A + A′C
		C=0	R = AND (A, B)	R = AB
	KMD Gate 4	B=C=1;D=0	Q = NOT(A)	Q = A′
		B=1	Q = NAND (A,C)	Q = A′ + AC′
		B=1; D=0 C=0	T = OR (A,C) Q = OR(A,C)	T = AC′ + C Q = AC′ + C

Table 2: Comparison of the proposed gates

Gates	Number of Cells used	Quantum Cost	Area (in µm2)
Fredkin gate [13]	187	5	0.19
DKG gate [13]	752	6	1.24
MRG GATE [13]	456	6	0.52
NFT gate[14]	128	-	0.142
KMD Gate 1	169	10	0.19
KMD Gate 2	121	10	0.13
KMD Gate 3	116	6	0.19
KMD Gate 4	244	12	0.42

operation in computer architecture design. A dedicated hardware module is incorporated into the division as part of the processor. It is a fundamental issue to identify the efficient division algorithm as per the IEEE 754 standard [27]. A complete division operation is a compound of sequential basic operations.

A proposed fault tolerant floating point division unit consists of the following elements: multiplexer, Parallel In Parallel Out (PIPO) left shift register, adder/subtractor unit, and rounding and normalization registers. All these functional units are designed as fault-tolerant.

There are two possible methods available for the division of integer numbers. They are restoring and non–restoring divisions. In both the methods, the base operations are addition/subtraction and shifting the variables [28].

For n bit D is a dividend (2n bits to store remainder and quotient after division), V is divisor, and Q is quotient register,

Restoring Division:
1. Shift the D (2n) to left one position.
2. Subtract V (n) from D(V-D) and place the result back in D.
3. If the Sign of D is 1, set MSB of Q_0 to 0 and add V back to D (Restore); otherwise, set Q_0 to 1.

Non-Restoring Division:
Step 1: (n times)
1. If the sign of D is 0, shift D left to the one-bit position and subtract V from D; otherwise, shift D and add V to D (V+D).
2. Now, if the sign of D is 0, set Q_0 to 1; otherwise, set Q_0 to 0.

Step 2:
1. If the sign of D is 1, add V to D (V+D).

In the restoring division, the left shift and subtract operation are equivalent to 2D-V. If D is negative, restore D and left shift, then subtract V which is equivalent to 2D+V. The latter case is used in the non-restoring division; which reduces the number of logical operations.

The following significant changes are made in the above algorithm and flow diagram which are shown in Fig.4:
- In the normal division, the dividend is shifted to the right side, here in the proposed method, it is left shifted.
- The shifting is advanced to the first place rather than after the subtraction.
- The remainder and quotient register are combined to form a single dividend register.

The complete flow diagram of restoring division is shown in Fig.4; in the case of non-restoring, and the path is slightly changed as combining 3a and 3b of Fig.4 without restoration. In addition, to meet the IEEE 754 standard of floating point representation, the rounding and normalization are carried out.

Algorithm:
Inputs: D (Dividend); V (Divisor) and Sel=0.
Outputs: R (Remainder) and Q (Quotient)

Steps:
1. Initial: Clk=High; SP=0; Sel=0; Count=0; D=0; V=0 (Registers are Initialized)
2. If (Clk)
   If (SP= =0)
   n-bit Inputs are parallel loaded in operand registers(D & V).
3. Else if (SP= =1 & Hold = = 0& Count <n)
   The operands are forwarded to n-bit parallel adder (as per non-restoring algorithm 2's complement addition).
   The output of the previous step is loaded into the D register and serial left shift one position in PIPO shift register.
4. If the partial result is positive (MSB=0) set Q_0=1; otherwise set Q_0=0;
5. Count = Count + 1;
6. If (Count>=n)
   If the result is negative; restore D and do rounding and normalization.
7. Dividend register MSB = Remainder; LSB = Quotient.
8. End.

5.1 Working Principle

The complete data path of fault-tolerant floating point division is shown in Fig.5. The significant units are multiplexers, registers (F2G), parallel adder, rounding and normalization units. The necessary control signals are

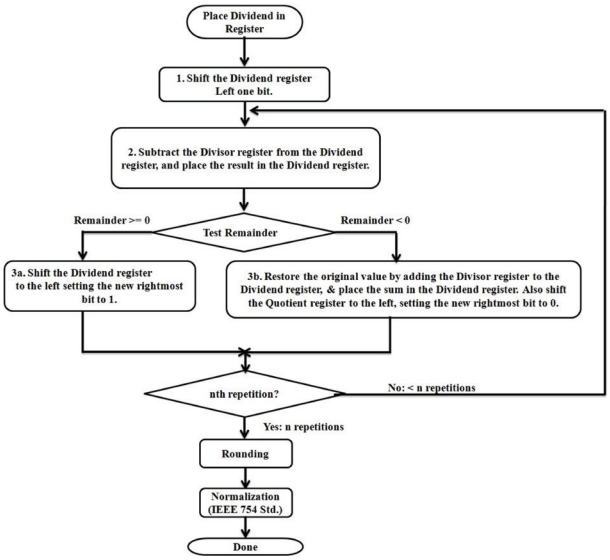


Figure4: Flow Diagram of floating point division

Clk, load, sel, SP, set q_0, shift and hold. These signals are released from the control unit at the appropriate time based on the Clk timing.

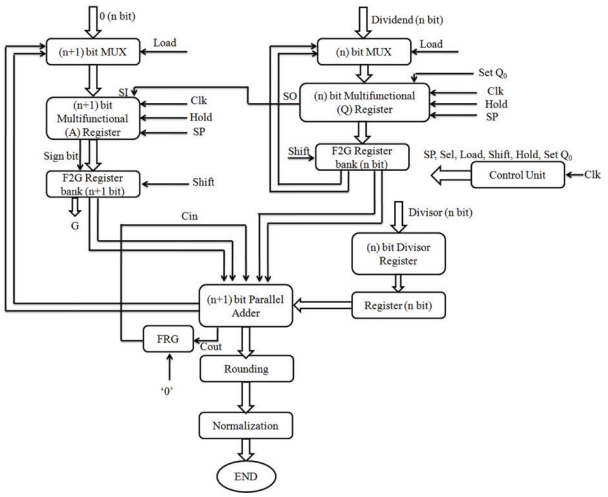


Figure 5: Fault-tolerant floating point division unit

When Clk is available and load & SP signals are high, the multifunctional registers are parallel loaded with '0' in 'A' register, and dividend in the Q register and divisor register are already loaded. In the next Clk, the dividend and divisor are loaded into (n+1) parallel adder. The adder performs the 2's complement addition, and the partial results are again stored back to higher order bits of the multifunctional register. In the meantime, left shifted (via F2G register) dividend is loaded back through the multiplexer and the MSB bit is serially shifted to the 'SO' of A register.

Here, non–restoring division is followed in order to reduce the number of computations [28]. So, it is not necessary to restore the dividend after an unsuccessful subtraction (ie. the partial result is negative). Instead, the partial result is 2's complemented via F2G register bank and an FRG register during the next cycle. The sign bit of the partial result decides the Q_0 value (ie. Q_0=0, if sign =1; 1, otherwise). The same procedure is repeated for 'n' Clk cycle. After n-cycle, if the result is negative, restoration takes place; otherwise, register Q contains the quotient and 'A' register contains the remainder of the successful division.

5.2 Key elements of reversible fault-tolerant division unit

The major functional units of reversible fault-tolerant division unit are multiplexers, operands registers, adder/subtractor, PIPO register, rounding and normalization registers. These functional units are being constructed using fault-tolerant gates; thereby the circuit becomes a fault-tolerant one.

Multiplexer:
KMD Gate 3 can be configured as a multiplexer with 'A' input as select line and others (B & C) are input data. The 3rd output provides the selected data. Fig.6 represents the 2 input multiplexer; when Sel=0; then C is selected; otherwise B is selected. The n-bit multiplexer can be derived by cascading the single bit structure. Here, the input D=0 provides 'sel' to be passed on to the next stage. The quantum cost of the n-bit Mux is 7n, and it has 2n garbage outputs.

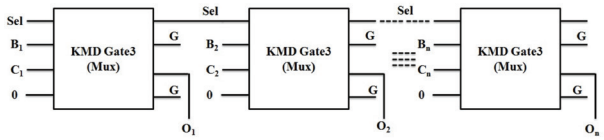


Figure 6: 2-input n-bit multiplexer

Reversible Parallel In Parallel Out (PIPO) Shift Register:
D latch can be derived from KMD Gate 3 as shown in Fig.7a. The 3rd output of the gate generates the necessary Q for the latch. A n-bit Parallel in Parallel Out (PIPO) is constructed by cascading the D-latches as shown in Fig.7a. The quantum cost of the n-bit PIPO is 7n, and it has 'n' garbage outputs.

A multifunctional register is designed using D-latch and multiplexers. It acts as a left shift register; PIPO register, SISO register, and normal storage register according to the control signal. Fig.7b shows the multifunctional register with control signals hold & SP and data signals SI (Serial Input) & PI (Parallel Input). The symbol of the register is shown in Fig. 7c. n-bit register has the quantum cost of 21n, number of gates 3n and 4n garbage outputs.

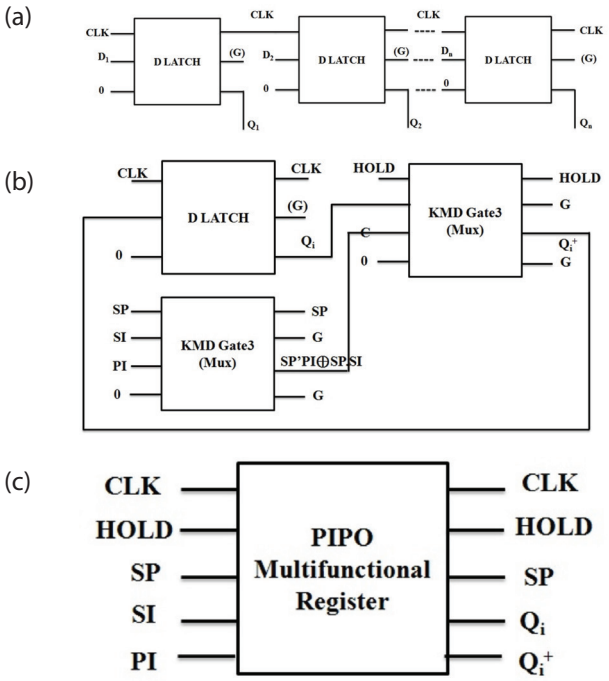


Figure 7: (a) n-bit PIPO Register (b) Construction of Multifunctional Register and (c) Symbol of Multifunctional Register

The output behavior of the multifunctional register is tabulated in Table 3 which is derived from equation (2). Here, the parallel load operation selects the external input, and the serial input transfers the previously computed data to the next register.

$$Q_i^+ = Hold'SP'I_i + Hold'SP.Q_{i-1} + Hold.Q_i \quad \ldots\ldots\ldots \quad (2)$$

The proposed D-latch using KMD Gate is having less of quantum cost, garbage output and gate count compared to the available latches as from Table 4. From the table, it is observed that the number of reversible gates used to construct D-latch in [8] is 7 which lead to high quantum cost and more garbage output, whereas the proposed design consists of one gate and least quantum cost of 9.

Table 3: Truth table for multifunctional register

Hold	SP	Qi+ (Next Output)
0	0	Ii (Input Loaded to Register in Parallel)
0	1	Qi-1 (Left Shift & LSB receiving input from Serial Input)
1	X	Qi (Maintaining Previous Value)

Fault Tolerant Reversible Adder:
The reversible fault-tolerant adder is being constructed using KMD Gate 4 with the quantum cost of 24 as shown in Fig.8. So, an n-bit adder consists of 21n quantum cost, 2n constant inputs, and 3n garbage outputs.

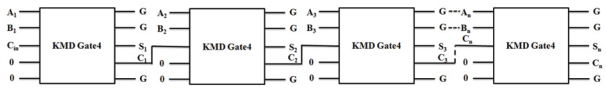


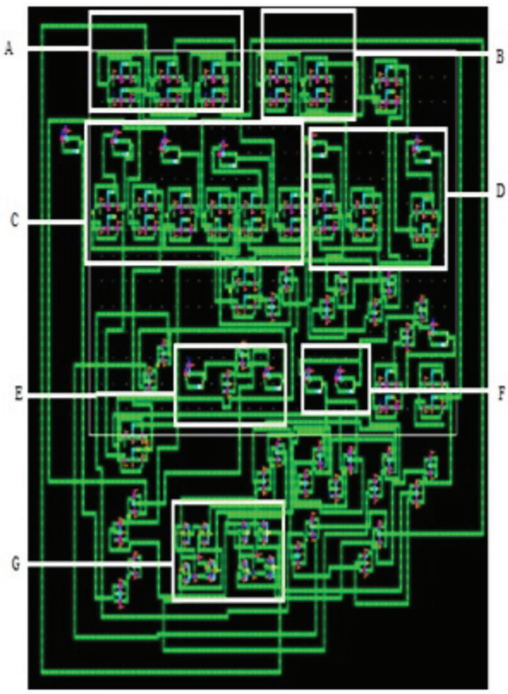
Figure 8: n-bit Fault-Tolerant Reversible Adder

Table 4: Performance analysis of D-latch

Parameters	[22]	[23]	[8]	Proposed design	Improvement % w.r.t		
					[22]	[23]	[8]
Quantum cost	14	10	47	9	35%	10%	80.8%
Garbage outputs	4	2	6	1	75%	50%	83.3%
Gate Count	3	3	7	1	66.6%	66.6%	85.7%
Number of Cells	-	-	-	116	-	-	-
Area	-	-	-	0.52μm²	-	-	-
Clock Zone	-	-	-	4	-	-	-

6 Results and Discussion

To perform the comparison between different reversible division logic circuits, few chosen parameters are,



A) (n+1)-bit Multiplexer
B) n-bit Multiplexer
C) (n+1)-bit Shift Register
D) n-bit Shift Register
E) Rounding Register
F) Normalization Register
G) Parallel Adder

Figure 9: 2-bit fault tolerant reversible divider circuit

quantum cost, garbage outputs, number of gates and constant inputs.

The QCA realization of the above functional units and entire division unit is done using QCADesigner 2.0.3 tool as shown in Fig.9. The structure consists of multiplexer, registers, adders, normalization and rounding off units. All those modules are integrated to form the fault-tolerant reversible floating point division unit.

Single bit multiplexers are combined to form the 2-bit and 3-bit multiplexers which receives dividend and zeros. At the same time, divisor register receives another divisor. The operands are then forwarded to the multifunctional register. This register is constructed using D-latch and multiplexer as in Fig.7. Then the operands are forwarded to the reversible adder. In the adder after every clk signal, the partial output is shifted one bit left. After n (number of bits) clk pulses, the adder output is forwarded to the rounding and normalization register to normalize the division as per the IEEE 754 single precision standard.

The cost and other parameters calculated for individual units of the n-bit division unit is shown in Table 5a. The major components are derived from KMD Gate 3, F2G, and KMD Gate 4. Since KMD Gate 3 is utilized to construct the multiplexer and multifunctional register, the uniformity of the divider is majorly improved.

Table 5b-e shows the comparison of quantum cost, number of gates, delay, and constant inputs of the division unit for 2 bits to 256 bits. From those tables, it is observed that the conventional and high-speed division array is exponentially increasing of Quantum Cost, Delay, and Constant Inputs with respect to a number of bits. But the proposed methodology is having linear relationship with the number of bits. So, proposed division unit can be utilized in any of the processor design. It is inferred that quantum cost, number of gates and delay are improved on a significant level compared to [25, 26]. For example, for the 256-bit division unit, the best available method has 3346 number of gates and delay [8]; while the proposed method has only 3076.

The pictorial representation of the consumption of quantum cost, number of gates, delay and constant inputs are shown in Fig. 10a-d. It is evident that the proposed method is having linear relation with number of bits, while the existing methods having quadratic relation with number of bits.

Table 5a: Performance measure calculations of individual modules of the division unit

S. No.	Module	Number of Bits	Gates Used	Number of Gates	Delay	Quantum Cost	Garbage Output	Constant Input
	Multiplexer	n	KMD Gate 3	n	n	6n	2n	n
		n+1		n+1	n+1	6n + 6	2n+2	n+1
	Multifunctional Register	n		3n	3n	18n	5n	3n
		n+1		3n+1	3n+1	18n+18	5n+5	3n+3
	Divisor Register	n	F2G	n	n	2n	2n	n
	Parallel Adder	n+1	KMD Gate 4	n+1	n+1	12n+12	3n+3	2n
	Register	n	F2G	n	n	3n	-	n
		n+1		n+1	n+1	3n+3	n+1	n+1
	Other Gates	1	Fredkin	1	1	5	2	1
	Total Cost			12n+4	12n+4	68n+44	20n+13	13n+6

Table 5b: Comparison of quantum cost for the existing and the proposed division unit.

Number Of Bits	Existing [25]		Existing [26]		Proposed
	Restoring	Non-Restoring	Conventional Division Array	High Speed Division Array	
2	210	203	33	82	180
4	360	353	74	165	316
8	660	653	204	415	588
16	1260	1253	656	1251	1132
32	2460	2453	2328	4267	2220
64	4860	4853	8744	15675	4396
128	9660	9653	33864	59995	8748
256	19360	19253	133256	234651	17452

Table 5c: Comparison of number of gates for the existing and the proposed division unit.

Number Of Bits	Existing [25]		Existing [26]		Existing [8]	Proposed
	Restoring	Non-Restoring	Conventional Division Array	High Speed Division Array		
2	59	57	13	34	44	28
4	95	93	33	69	70	52
8	167	165	79	175	122	100
16	311	309	251	531	226	196
32	599	597	883	1819	434	388
64	1172	1173	3299	6699	850	772
128	2327	2325	12739	25675	1682	1540
256	4631	4629	50051	100491	3346	3076

Table 5d: Comparison of delay for the existing and the proposed division unit.

Number Of Bits	Existing [25]		Existing [26]		Existing [8]	Proposed
	Restoring	Non-Restoring	Conventional Division Array	High Speed Division Array		
2	54	52	24	12	43	27
4	88	86	54	27	69	51
8	156	154	150	85	121	99
16	292	290	486	297	225	195
32	564	562	1734	-	433	387
64	1108	1106	6534	-	849	771
128	2196	2194	25350	-	1681	1539
256	4372	4370	99846	-	3345	3075

Table 5e: Comparison of constant inputs for the existing and the proposed division unit.

Number Of Bits	Existing [25]		Existing [26]		Proposed
	Restoring	Non-Restoring	Conventional Division Array	High Speed Division Array	
2	36	34	5	20	32
4	58	56	13	39	58
8	102	100	41	95	110
16	190	188	145	279	214
32	366	364	545	935	422
64	718	716	2113	3399	838
128	1408	1406	8321	12935	1670
256	2830	2828	33025	50439	3334

Table 6: Comparison of performance measures for n-bit division unit

Parameters	Existing [25]		Existing [26]		Proposed
	Restoring	Non-Restoring	Conventional Division Array	High Speed Division Array	
No. of Gates	18n+23	18n+21	3(n+2)2+2n/4	(n+2)(3n+11)/2	12n+4
Delay	17n+20	17n+18	3(n+2)2/2	-	12n+3
Garbage outputs	12n+18	12n+16	(n+2)2/2	(n+2)(3n+22)/4	20n+13
Quantum Cost	75n+60	75n+53	4(n+2)2+n/2	(n+2)(7n+27)/2	68n+44
Constant Input	11n+14	11n+12	(n+1)2+1/2	(n+2)(3n+14)/4	13n+6

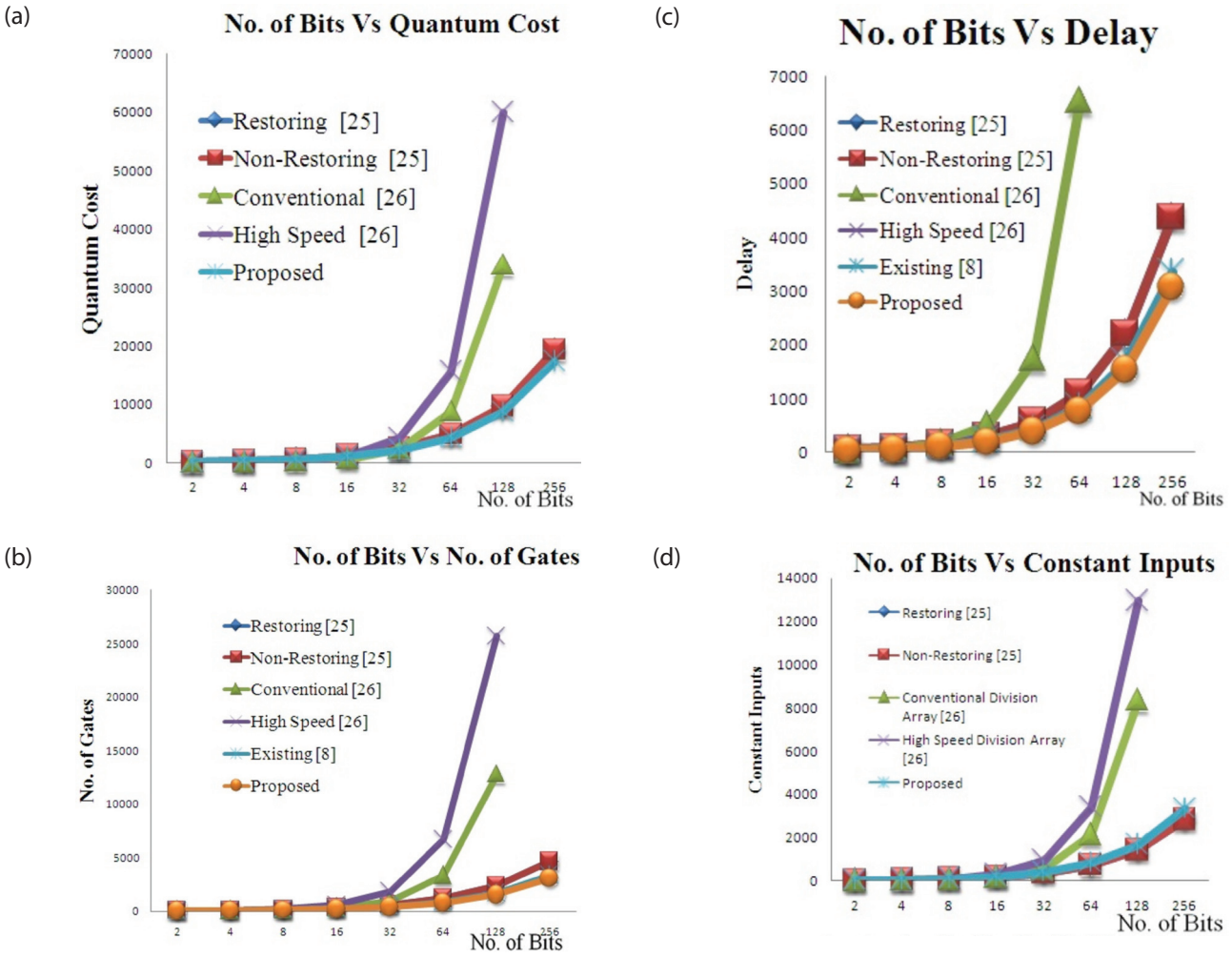


Figure 10: a. Number of bits vs Quantum cost; b. Number of bits vs Number of gates; c. Number of bits vs Delay; d. Number of bits vs Constant inputs

The estimated performance measurement of n-bit division unit is tabulated in Table 6. It is observed that the dependency factor 'n' – the number of bits is greatly reduced by the proposed method with respect to the existing [25, 26]. For example, conventional and high-speed division array have exponential relation with 'n', whereas the proposed method has a linear relationship. Moreover, the additional constant cost involved in 'n' is also reduced for the proposed method. From the last column of Table 6, it is observed that, the worst case additional cost is 44, but in the existing method, it is up to 60.

7 Conclusion

In this paper, we have proposed a new n-bit fault-tolerant reversible floating point division unit (FTRFPD) which functions according to the non-restoring algorithm. The proposed division unit is being constructed by the fault-tolerant reversible KMD Gates. The number of cycles and the number of computations required to complete the division is drastically reduced. Here, KMD Gate 3 is utilized to construct multiplexer and multifunctional register, for maintaining uniformity in the entire circuit design. The comparative results prove that the proposed method has greater improvement in the number of gates (12n+4), delay (12n+3) and quantum cost (68n+44) with respect to the existing methodology. Also, it has significant improvement in garbage output and constant input. The entire work is functionally verified using QCADesigner 2.0.3 tool. Furthermore, the efficient division unit can be incorporated into any ALU for floating point operation.

8 References

1. R. Landauer, "Irreversibility and heat generation in the computing process", *IBM J. Research and Development*, Vol. 5(3), July 1961, pp:183-191.

2. C.H.Bennett, "Logical reversibility of computation", *IBM J. Research and Development*, Vol. 17, November 1973, pp:525 – 532.
3. Chris Thachuk, "Logically and Physically Reversible Natural Computing": *Proc. 5th International Conference on Reversible Computation (RC'13)*, Vol. 7948,2013, pp. 247–262.
4. Behrooz Parhami, "Fault-Tolerant Reversible Circuits", *Proc. 40th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, October 2006, pp. 1726–1729.
5. K. Sridharan, Vikramkumar Pudi, "Design of Arithmetic Circuits in Quantum Dot Cellular Automata Nanotechnology", *Studies in Computational Intelligence*, Springer, Vol. 599, 2015.
6. Craig S Lent, Mo Liu and Yuhui Lu, "Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling", *Nanotechnology*, Vol. 17, 2006, pp.4240-4251.
7. Biswas, Md.Mahmudul Hasan, Chowdhury, Md.Hasan Babu, "Efficient approaches for designing reversible Binary Coded Decimal adders", *Microelectronics Journal*, Vol. 39, 2008, pp. 1693–1703.
8. Hafiz Md. HasanBabu, Md. Solaiman Mia, "Design of a compact reversible fault tolerant division circuit", *Microelectronics Journal*, Vol. 51, 2016, pp.:15–29.
9. T. Toffoli,"Reversible Computing", *Technical Report* MIT/LCS/TM-151, 1980, pp. 1-37.
10. E. Fredkin, T. Toffoli, "Conservative logic", *Int. J. Theor. Phys*., Vol. 21,1980, pp.219–253.
11. A. Peres, "Reversible logic and quantum computers", *Phys. Rev*., Vol. 32, 1985, pp. 3266–3276.
12. R. Feynman, "Quantum mechanical computers", *Found. Phys*., Vol. 16 (6), 1986, pp. 11-20.
13. Papiya Biswas, Namit Gupta, Nilesh Patidar, "Basic Reversible Logic Gates and It's QCA Implementation", *Int. Journal of Engineering Research and Applications*, Vol. 4 (6), June 2014, pp.12-16.
14. Ali NewazBahar, Sajjad Waheed, Nazir Hossain, "A new approach of presenting reversible logic gate in nano-scale", DOI 10.1186/s40064-015-0928-4, Springer, 2015.
15. Kalyan S. Perumalla, "Introduction to Reversible Computing", *CRC Press*, 2014.
16. Kamaraj, A., P.Marichamy, S.Karthika Devi, and M.Nagalakshmisubraja. "Design and Implementation of Adders using Novel Reversible Gates in Quantum Cellular Automata", *Indian Journal of Science and Technology*, Vol 9(8), 2017, pp. 1-7.
17. MD. Saiful Islam, Zerina Begum, "Reversible Logic Synthesis of Fault Tolerant Carry Skip BCD Adder", *Journal of Bangladesh Academy of Sciences*, Vol. 32 (2), 2008, pp.243-249.
18. Sen, Ganeriwal, Sikdar, "Reversible Logic-Based Fault-Tolerant Nanocircuits in QCA", *Hindawi*, dx.doi.org/10.1155/2013/850267, May 2013, 9 pages.
19. Valinataj, Mirshekar, Hamid Jazayeri, "Novel low-cost and fault-tolerant reversible logic adders", *Computers and Electrical Engineering,* Vol. 53, 2016, pp.56–72.
20. Md. Riazur Rahman, "Cost Efficient Fault Tolerant Decoder in Reversible Logic Synthesis", *International Journal of Computer Applications,* Vol. 108 (2), December 2014, pp. 7-12.
21. Xuemei, Fulong, Liangmin, Yonglong, Min Hu, "Design of fast fault tolerant reversible signed multiplier", *International Journal of the Physical Sciences,* Vol. 7(17), 23 April 2012, pp. 2506 - 2514.
22. A.V. Anantha Lakshmi, Gnanou Florence Sudha, "Design of a reversible floating-point square root using modified non-restoring algorithm", *Microprocessors and Microsystems*Vol. 50, 2017, pp. 39–53.
23. Belayet Ali, Mosharof Hossin, Eneyat Ullah, "Design of Reversible Sequential Circuit Using Reversible Logic Synthesis", *International Journal of VLSI design & Communication Systems (VLSICS)* Vol.2 (4), December 2011, pp. 37- 45.
24. Thapliyal H, Vinod A.P., "Design of reversible sequential elements with the feasibility of transistor implementation", *In Proceedings of the IEEE International Symposium on Circuits and Systems*, 2007, pp. 625–628.
25. Faraz Dastan, Majid Haghparast, "A novel nanometric fault tolerant reversible divider", *International Journal of the Physical Sciences* Vol. 6(24), 16 October 2011, pp. 5671-5681.
26. L.Jamal, H.M.H.Babu, "Efficient approaches to design a reversible floating point divider", *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems*, Vol. 13, 2013, pp.3004–3007.
27. IEEE standard for floating-point arithmetic, IEEE Std.754-2008, 2008, pp.1–58.
28. Carl Hamacher, Vranesic, Safwat Zaky, "Computer Organization", 5th Edition, Tata McGraw-Hill, 2011.

Arrived: 09. 04. 2018
Accepted: 13. 07. 2018