

A Novel Nanometric Reversible Signed Divider with Overflow Checking Capability

¹Faraz Dastan and ²Majid Haghparast

¹Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

²Department of Computer Engineering, Shahre-Rey Branch,
Islamic Azad University, Tehran, Iran

Abstract: One of the best approaches for designing future computers is that we use reversible logic. Reversible logic circuits have lower power consumption than the common circuits, used in computers nowadays. In this study we propose a new reversible division circuit. This reversible division circuit is signed divider and has an overflow checking capability. Among the designed and proposed reversible division circuits, our proposed division circuit is the first reversible signed divider with overflow checking capability which has been designed. In this circuit we use some reversible components like reversible parallel adder, reversible left-shift register, reversible multiplexer, reversible n-bit register and reversible n-bit register with parallel load line. In this paper all the scales are in the nanometric area.

Key words: Nanometric circuits, nanotechnology, overflow, quantum computing, reversible logic, reversible signed divider

INTRODUCTION

In designing process of logic circuits, energy consumption is an important factor. In Ordinary irreversible logic circuits due to information loss, we have energy dissipation. Landauer in his research said that the amount of energy that is dissipated for every irreversible bit operation is at least “ $KT \ln 2$ ” joules, where $K = 1.3806505 \times 10^{-23} \text{ m}^2 \text{ kg}^{-1} \text{ K}^{-1}$ is a Boltzmann’s constant and T is temperature in which operation is performing (Landauer, 1961; Parhami, 2006). Reversible logic has been developed to deal with this problem. In Reversible logic circuits inputs can be obtained from outputs and in reversible logic, $KT \ln 2$ energy is not wasted (Bennett, 1973; Hayes, 2006). Important characteristic of reversible circuits is that the number of inputs and numbers of outputs are equal. Error management in reversible logic circuits is easier than the common circuits. Reversible logic is good choice for optical computing, quantum computing and nanotechnology based systems. Quantum computing without using reversible logic circuits is not easily accessible. In reversible circuits fan-out is not allowed (Parhami, 2006; Perkowski *et al.*, 2001; Haghparast and Navi, 2008). Reversible circuits are formed of reversible gates (Vasudevan *et al.*, 2004). Arithmetic units are important units in computer hardware because they have many uses in computer systems. One of these arithmetic units is division unit. Designing best reversible arithmetic circuits is very challenging. There is one proposed reversible division circuit in previous

papers (Nayeem *et al.*, 2009) but it is unsigned division circuit and it has not overflow checking capability.

In this study we propose a new reversible division circuit. This proposed divider is signed division hardware and has an overflow checking capability. Our proposed reversible division circuit is the first reversible signed division circuit with overflow checking which has been designed. Our proposed reversible divider composed of reversible components like reversible multiplexer, reversible PIPO left-shift register, reversible register, reversible register with parallel load line and reversible parallel adder. It is to be noted that all the scales are in the nanometric area.

METHODOLOGY

Basic concepts: In this section some background information is provided. These are information about reversible gates.

Definitions:

Reversible logic: If there is one to one correspondence between inputs and outputs in function ‘ K ’ (K has “ n ” input and “ n ” output), ‘ K ’ is reversible. Thus the inputs vector is uniquely, determinable from outputs vector (Haghparast *et al.*, 2008).

Garbage output: If the output of a gate is used nowhere of a circuit, this output called garbage output.

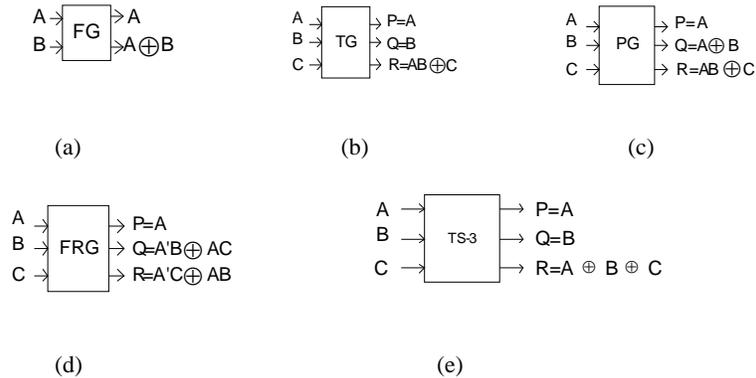


Fig. 1: Some useful reversible logic gates, (a): Feynman Gate, (b): Toffoli Gate, (c): Peres Gate, (d): Fredkin gate, e: TS-3 Gate

Constant inputs: for make an $m \times n$ function ‘K’ reversible, some inputs are added to it. These inputs called constant inputs (Haghparast *et al.*, 2009). In reversible circuits constant inputs refer to inputs that is permanently ‘1’ or ‘0’.

Quantum cost: Number of 1×1 or 2×2 reversible logic gates, needed to make the reversible gate because the quantum gates larger than 2×2 are not directly realizable in the quantum technology (Barenco *et al.*, 1995; Kaye *et al.*, 2007; Haghparast *et al.*, 2009; Mohammadi *et al.*, 2009).

Reversible logic gates: There are a lot of reversible logic gates that have been proposed in previous papers. Some of these gates are presented in this section. Feynman Gate (FG) (Feynman, 1985) (Fig. 1a), Toffoli gate (TG) (Fig. 1b) (Toffoli, 1980), Peres Gate (PG) (Peres, 1985) (Fig. 1c), Fredkin Gate (FRG) (Fredkin and Toffoli, 1982) (Fig. 1d) and TS-3 (Thapliyal *et al.*, 2006) (Fig. 1e) are very useful gates.

Division approaches in computer systems: For binary logic computer systems, special algorithms have been proposed to perform arithmetic tasks like multiplication and division. Multiplication in computer systems is done by repeated additions and consequently, division is done by repeated subtraction. Common algorithm, used for division in computer systems is shift-subtract algorithm (Parhami, 2000). In every division process, inputs are dividend (D) and divisor (d) and outputs of the division process are quotient (q) and remainder (s). Shift-subtract algorithm, performed in multi steps or multi cycles (Parhami, 2000). In every step i , dividend (D) is shifted one bit to the left and then divisor subtracted from dividend. If result of this subtract operation is positive, we insert ‘1’ to the q_i . Otherwise if result of the subtract operation was negative, ‘0’ is inserted to the q_i ; in this

condition, the partial remainder must be restored. Restoration means that, divisor adds to the partial remainder. Division operation in computer systems can be done by two approach, restoring approach and non-restoring approach. In restoring approach if result of subtract operation is negative, restoring phase is performing immediately by adding a divisor to the partial remainder. In non-restoring approach, if result of the subtract operation is negative, restoring approach is done after the left-shift operation (Nayeem *et al.*, 2009; Hayes, 1998; Parhami, 2000). We consider that number of dividend bits is double of the divisor bits. For example dividend has $2n$ -bit and divisor has n -bit, thus in the system that has an n -bit memory words, two memory words are used to store the dividend and one memory word is used to store the divisor. Left-shift and then subtract or add operation form one step or one cycle. When steps of the division process reach to the end (if divisor has n -bit we have n steps or n cycles for division process), if partial remainder is negative, restoring must be done to have correct value of final remainder. Restoration is easily done by adding the divisor to the partial remainder (Nayeem *et al.*, 2009) (Hayes, 1998).

In division process we may have an overflow. When the division operation is done by hardware, the overflow is a significant problem. In the computer systems the length of registers in the processor is finite and registers can hold values with certain number of bits. For example about divide overflow, consider that we have processor with n -bit registers. In division operation we use one register to hold the divisor and two register to hold the dividend. The result bits of divide operation (quotient) will be stored in n -bit register too. In the case that the divide operation will result $(n+1)$ -bit for quotient, we have an overflow condition. Overflow condition occur when high order half of the dividend bits is greater than or equal with the divisor (Mano, 1993). For example high order half of the dividend bits are 1010 and the divisor’s bits are

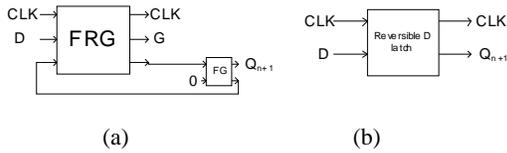


Fig. 2: Reversible D latch (Thapliyal and Zvolinski, 2006) (a) Implementation of reversible D latch with FRG and FG; (b) Block diagram of reversible D latch

1000. Overflow condition must be detected. Detection can be done by computer hardware or software or combination of hardware and software (Mano, 1993).

With overflow checking, we can prevent division by zero conditions too (Mano, 1993). When overflow occur, special flip-flop is set. This flip-flop is called divide-overflow flip-flop (Mano, 1993). When divide-overflow high order half of the dividend bits is greater than or equal with the divisor (Mano, 1993). For example high order process begins, the divisor, subtract from high order half of the dividend. If output carry of subtract operation is '1', overflow condition will occur (Mano, 1993).

We consider that in n-bit memory words, one bit is used to hold a sign and thus magnitude consists of (n-1)-bit (Signed-magnitude representation) (Mano, 1993). To determine the sign of the quotient, sign bit of the dividend and divisor are used. We XOR (Exclusive OR) the sign

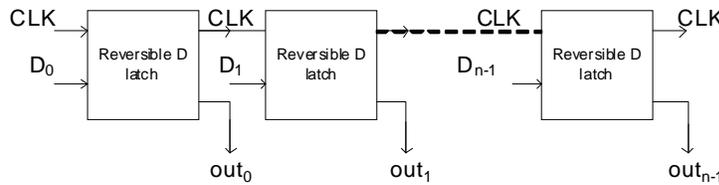


Fig. 3: Reversible n-bit register (Thapliyal and Zvolinski, 2006)

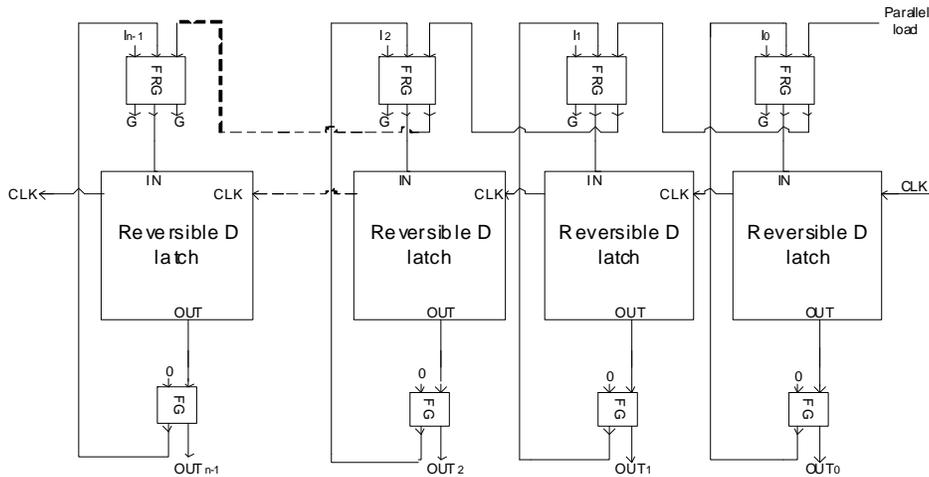


Fig. 4: Reversible n-bit register with parallel load line

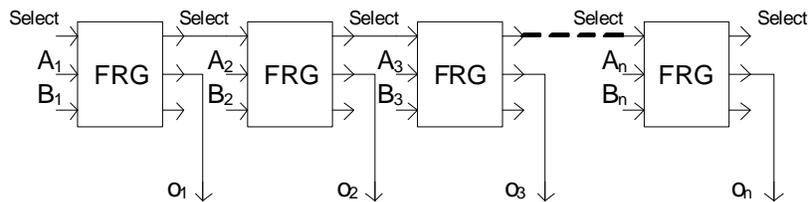


Fig. 5: Reversible two input n-bit MUX (Nayem et al., 2009)

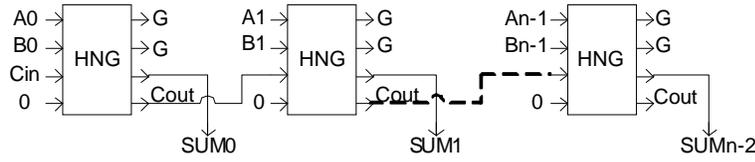


Fig. 6: Reversible (n-1)-bit parallel adder (Haghparast and Navi, 2008)

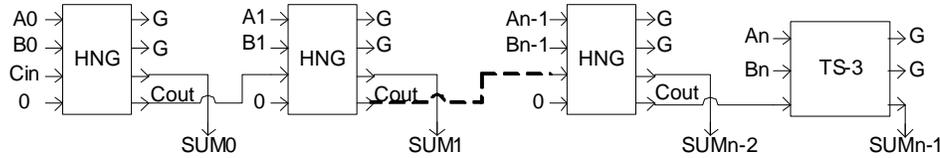


Fig. 7: Reversible n-bit parallel adder with ignoring output carry

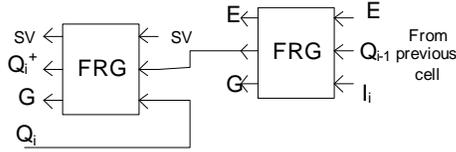


Fig. 8: Realization of function 1 with FRG gates (Nayeem et al., 2009)

bit of the dividend and sign bit of the divisor to obtain the sign bit of quotient and store it in the quotient's sign flip-flop (Q_s) (Mano, 1993). Remainder's sign is equal with the dividend's sign (Mano, 1993).

COMPONENTS ARE REQUIRED FOR REVERSIBLE SIGNED DIVIDER WITH OVERFLOW CHECKING CAPABILITY

Reversible register and reversible register with parallel load line: Reversible register is composed of reversible D latches (Thapliyal and Zwolinski, 2006). In Fig. 2, reversible D latch is shown. Figure 3 shows an n-bit reversible register (Thapliyal and Zwolinski, 2006).

Reversible register with parallel load line is presented in Fig. 4. This register is implemented with reversible D latches and FRG gates.

When parallel load line is '0', value of the register does not change and if parallel load line is '1', during clock pulse, parallel input can load into the register and appears in the output.

Reversible multiplexer: Reversible multiplexer is shown in Fig. 5. Reversible multiplexer (MUX) is realized with FRG gates (Nayeem et al., 2009). This MUX is two inputs and has one select line. According to Fig. 5, functionality of a MUX is shown in Table 1.

Table 1: Functionality of two-input MUX

Select	Output
0	A
1	B

Table 2: 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	\times	Q_i (No change)

Reversible parallel adder: Parallel adder is one of the mostly used circuits in computer systems. Figure 6 illustrates an (n-1)-nit reversible parallel adder. HNG gates are used to implement this parallel adder (Haghparast and Navi, 2008). In Fig. 7, n-bit parallel adder is illustrated. In n-bit parallel adder that realized with HNG gates and one TS-s gate, the output carry of add or subtract operation is ignored and the last bit of the result is obtained with two exclusive or (XOR) operation (Nayeem et al., 2009).

Reversible PIPO left-shift register: In Parallel Input-Parallel Output (PIPO) left shift register, data bits can load into the register in parallel. This register can shift left the data bits and also can hold its value for every time we want (Nayeem et al., 2009). These properties of PIPO left-shift register have many uses in computer circuits. In division circuit, PIPO left-shift register is used and it is one of the important units in reversible division circuit. To determine the shift register's action, some control lines are needed. Control lines, determine the left-shift register what to do. Table 2 shows control lines of the left-shift register (SV and E). When SV and E are 0, the left-shift register, shifts the data bits to the left, when SV is 0 and E is 1, data bits (I_i) can load into the left-shift register in parallel, when SV is 1, current value of the register is

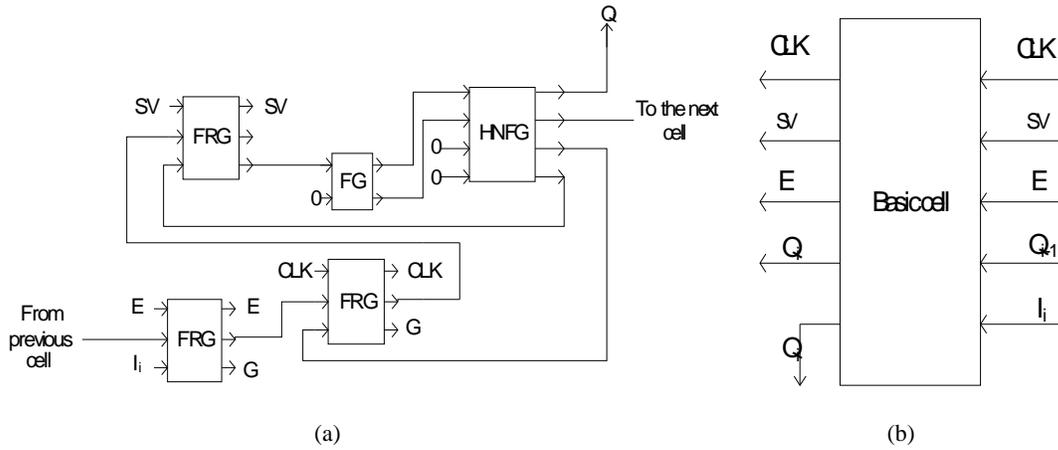


Fig. 9: (a) Basic cell of reversible PIPO left-shift register (Nayeem *et al.*, 2009), (b) Block diagram of reversible PIPO left-shift register's basic cell (Nayeem *et al.*, 2009)

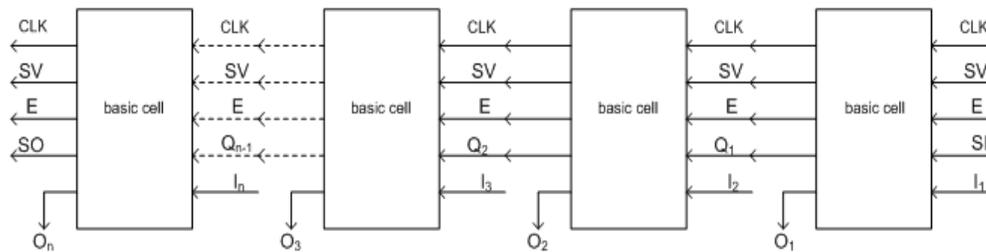


Fig. 10: Reversible PIPO left-shift register (n-bit) (Nayeem *et al.*, 2009)

saved. According to the Table 2, control function of left-shift register is

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

(Nayeem *et al.*, 2009). In Fig. 8, implementation of function 1 with FRG gates is illustrated. This control unit is used to form the PIPO left-shift register's basic cell (Fig. 9). Figure 10 presents n-bit PIPO left-shift register.

Our proposed reversible divider: At this section our proposed reversible signed divider with overflow checking capability is introduced. In previous sections, components that are needed for designing the reversible division circuit were presented. Figure 11 illustrates the proposed division circuit. In this division circuits we have two reversible parallel adder-subtractors (one (n-1)-bit and one (n)-bit), also there are three two-input MUXs (one n-bit and one (n-1)-bit and one FRG gate as a one bit MUX), two PIPO left-shift registers (one n-bit and one (n-1)-bit), four n-bit registers that three of them have parallel load line for controlling their inputs and three D latches. MSB bit of each register is used to hold the sign

bit. For preventing fan-out FG gates are used, also FG gates are used as a NOT gates. One Peres gate is used as a AND gate. In the beginning of the division operation, overflow condition must be checked. According to Fig. 11, at first clock, high-order half of (2n-2)-bit dividend (high-order half of dividend is 'A' and low-order half of dividend is 'B') and 1's complement of (n-1)-bit divisor are loaded into the (n-1)-bit adder-subtractor. Input carry of adder is '1' and thus subtract operation is done. Output carry of subtract operation determines that the overflow condition will appear in division process or not. If output carry is '1', there is overflow condition; therefore '1' is loaded into the overflow latch and computer will stop (division stop (Mano, 1993)). Output carry of the subtract operation also, connected to the parallel load line of Three n-bit registers. When there is no overflow condition (output carry of the subtract operation is '0') parallel load line of registers will be '1' and two half of dividend (A and B) and divisor will be loaded into the three registers and at the next clock they appear in the output of the registers. Therefore, sign bit of the dividend and sign bit of the divisor can be loaded into FG gate (operate as a XOR gate) to determine the

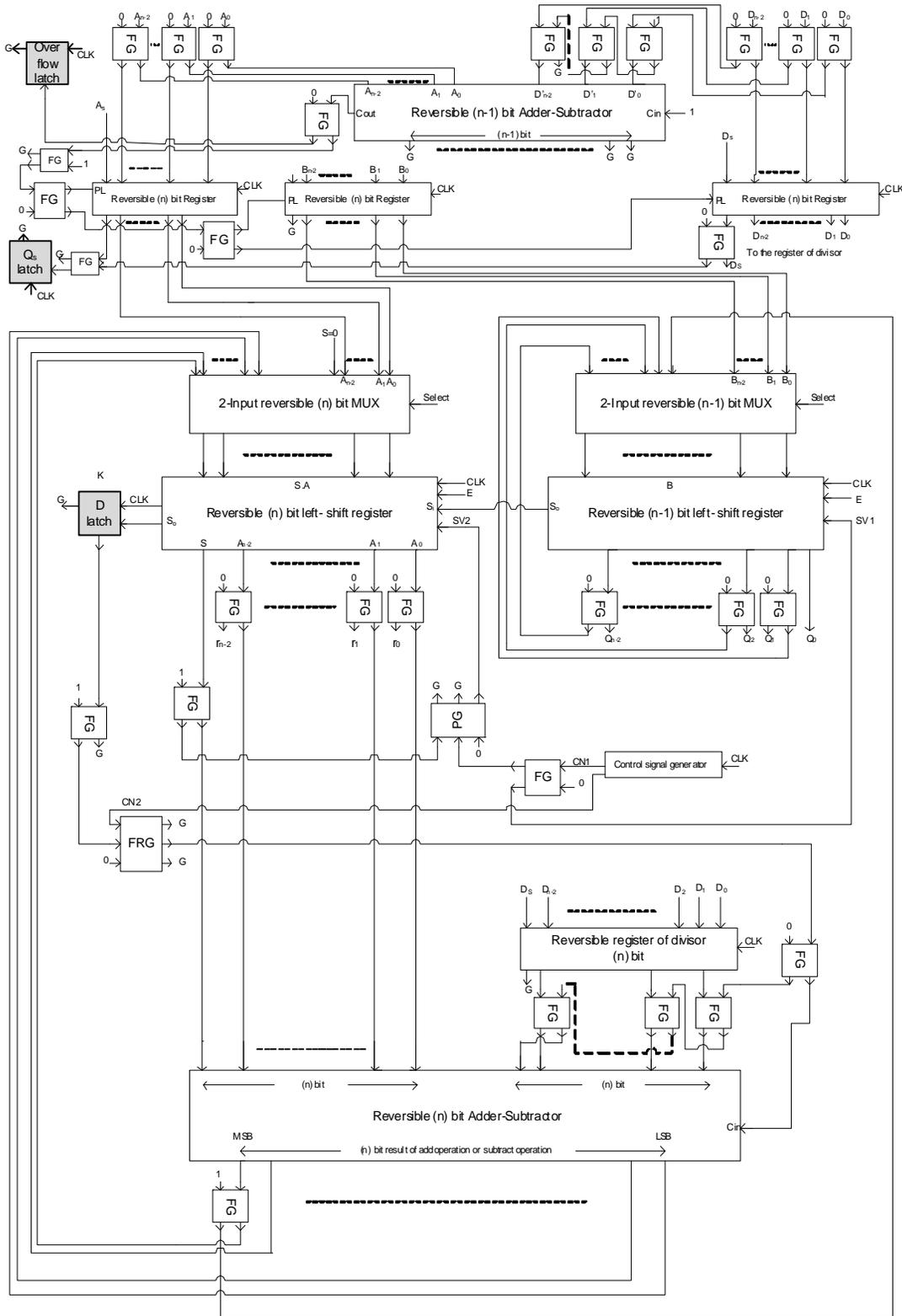


Fig. 11: Our proposed reversible division circuit

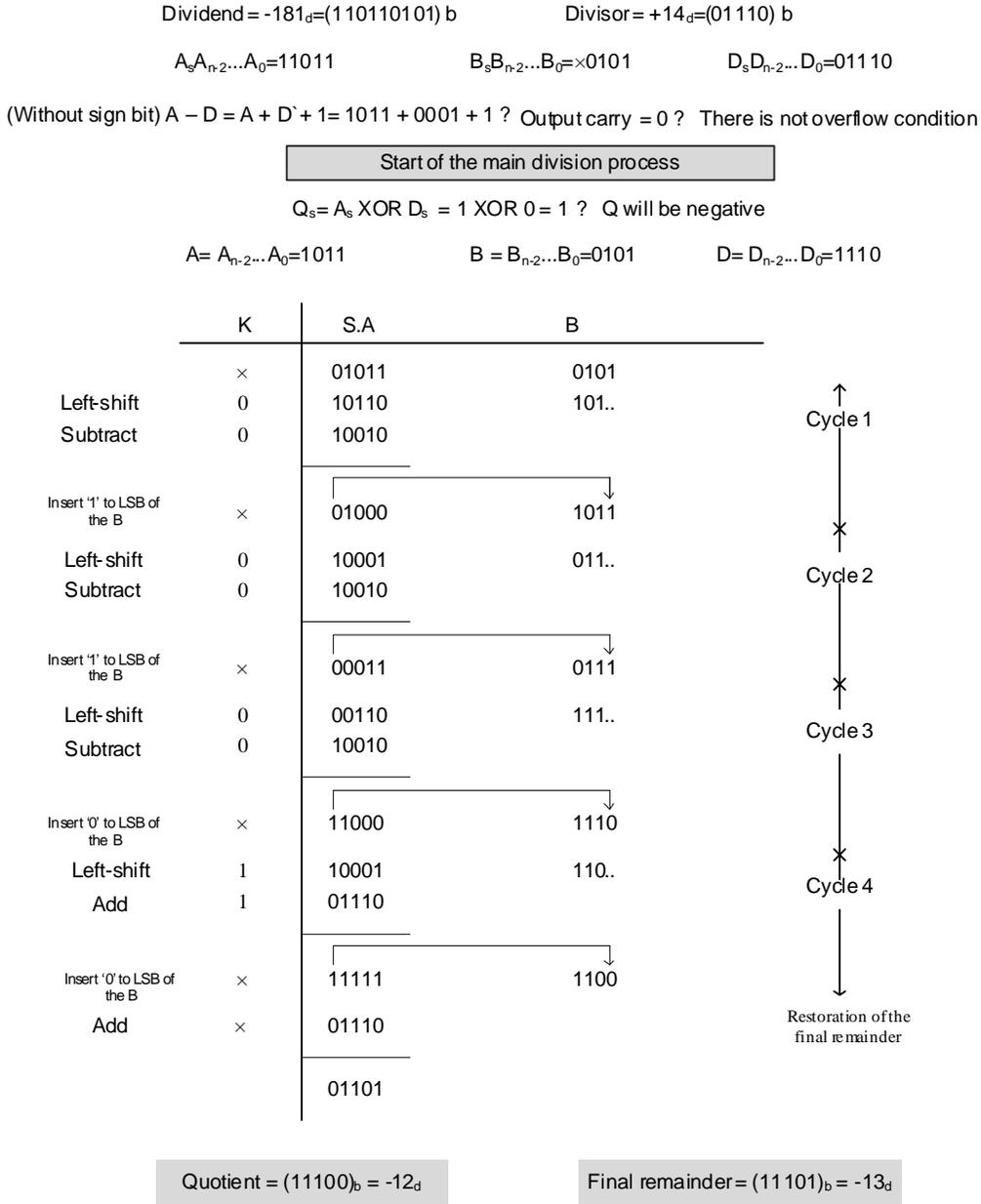


Fig. 12: Numerical example of our proposed reversible division circuit functionality

quotient's sign. Flip-flop that named Q_s holds the quotient's sign bit. After checking the overflow condition and determination of the quotient's sign, division process can be started. Like division process proposed in (Nayem *et al.*, 2009), during clock, when select lines of multiplexers are '0', $S = 0$ and initial value of A (high-order half of dividend), in the case that $E = 1$ and $SV2 = 0$, can be loaded into the S.A left-shift register and at the other MUX ((n-1)-bit MUX) initial value of B (low-order half of dividend), in the case that $E = 1$ and $SV1 = 0$, can be loaded into the B left-shift register. Control signal

generator, generates two control signals in its output (CN1 and CN2) and they are initially '0'. After that, select line is changed to '1', E is changed to '0', and left-shift operation is done at the shift registers. MSB bit of the (n-1)-bit left-shift register (B) is entered to the LSB bit of n-bit left-shift register (S.A) and value of S is shifted to the D latch (K). K determines that n-bit parallel adder-subtractor what to do. If $k = 0$, subtract operation (S.A-D) is done and adder-subtractor performs adding operation (S.A+D) if $K = 1$. Adding operation is needed for restoration of the partial remainder (when $S = 1$). At

the next clock, after add or subtract operation, MSB bit of the result enters to the FG gate (operate as a NOT gate) and complement of it, is loaded into the LSB bit of the (n-1)-bit left-shift register (B) to form the MSB bit of the quotient. Left-shift and then subtract or add operation form one cycle. At the end of the n-1 cycles, we have final remainder in the n-bit left-shift register (S.A) and quotient in the (n-1)-bit left-shift register (B). Control signal generator, changes the CN1 to '1' to final remainder and quotient can be saved in the left-shift registers. Peres gate does a 'S'.CN1' operation and SV2 is a product of this operation. If value of the S is '1' (after the n-1 cycles), the final remainder must be restored to correct value of the final remainder stored in the S.A shift register; therefore, E is changed to '1' and control signal generator changes the CN2 (select line of the one bit MUX) to '1' and then '0' can be loaded into the input carry of the n-bit parallel adder-subtractor. Adding operation (S.A+D) is done and then S will be '0' and thereby, SV2 (S'.CN1) will be '1' and thus in the next clock, correct value of the final remainder is saved in A (Nayeem *et al.*, 2009). Final remainder's sign and dividend's sign are equal (A_s).

Figure 12 illustrates a numerical example of the proposed division circuit's functionality. We consider that the computer system has 5-bit registers and MSB bit of each register determines the sign (magnitude in each register is 4-bit).

Evaluation of the proposed reversible division circuit:

In this section, we evaluate the proposed division circuit. At first, every components of this divider are evaluated.

- N-bit reversible MUX: no. of garbage outputs = n, quantum cost = 5n , no. of constant inputs = 1
- (n-1)-bit reversible MUX: no. of garbage outputs = n-1, quantum cost = 5n-5
- Three n-bit reversible registers with parallel load line: garbage outputs = 3*(2n+2)+1, quantum cost = 3*(12n), no. of constant inputs = 3*(2n)+1
- n-bit reversible register: garbage outputs = n+2, quantum cost = 6n, no. of constant inputs = n
- n-bit reversible PIPO left-shift register: no. of garbage outputs = 3n+2, quantum cost = 18n, no. of constant inputs = 3n
- (n-1)-bit reversible PIPO left-shift register: no. of garbage outputs = 3n, quantum cost = 18n-18, no. of constant inputs = 3n-3
- (n-1)-bit reversible parallel adder: no. of garbage outputs = 3n-3, quantum cost = 6n-6, no. of constant inputs = n
- n-bit reversible parallel adder with ignoring output carry: no. of garbage outputs = 2n, quantum cost = 6n-4, no. of constant inputs = n-1
- Three D latches: no. of garbage outputs = 3*2, quantum cost = 3*6, no. of constant inputs = 3*1

Table 3: Characteristic of the proposed reversible divider

Quantum cost	Garbage output	Constant input
106n-2	20n+21	19n+9

- Other gates: (no. FG gate = 6n+4, no. FRG gate = 1, no. Peres gate = 1) no. of garbage outputs = 8, quantum cost = 6n+13, no. of constant inputs = 4n+8

Table 3 shows a characteristic of the proposed reversible divider.

CONCLUSION

All of the computer users want to have computers that have low energy consumption and also very fast. To designing future computers, ordinary technologies are not suitable. Reversible logic can be a best choice to designing future computers. Fast computer needs a circuit that well-designed, even if we use new technologies like reversible circuits. Arithmetic circuits have an important role in computer systems. In this study we propose a new reversible signed division circuit that has an overflow checking capability. According to our knowledge, this reversible divider is the first proposed signed division circuit with overflow checking capability. All the circuits have nanometric scales.

REFERENCES

Barenco, A., C.H. Bennett, R. Cleve, D.P. Di-Vincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin and H. Weinfurter, 1995. Elementary gates for quantum computation. *Phys. Rev. A*, 52(5): 3457-3467.

Bennett, C.H., 1973. Logical reversibility of computation. *IBM J. Res. Dev.*, 17: 525-532.

Feynman, R., 1985. Quantum mechanical computers. *Optics News*, 11: 11-20.

Fredkin, E. and T. Toffoli, 1982, Conservative logic. *Int. J. Theor. Phys.*, 21: 219-253.

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

Haghparsat, M., S.J. Jassbi, K. Navi and O. Hashemipour, 2008. Design of a novel reversible multiplier circuit using HNG gate in nanotechnology. *World Appl. Sci. J.*, 3(6): 974-978.

Haghparsat, M., M. Mohammad, K. Navi and M. Eshghi, 2009. Optimized reversible multiplier circuit. *J. Circuits Syst. Comp.*, 18(2): 311-323.

Hayes, J.P., 1998. *Computer Architecture and Organization*. 3rd Edn., McGraw-Hill, New York.

Hayes, B., 2006. Reverse engineering. *Am. Sci.* 94: 107-111.

Kaye, P., R. Laflamme and M. Mosca, 2007. *An Introduction to Quantum Computing*. Oxford University Press, eBook-LinG, ISBN: 0-19-857000-7.

- Landauer, R., 1961. Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.*, 5: 183-191.
- Mano, M.M., 1993. *Computer System Architecture*. 3rd Edn., Books4Least, CA, USA, ISBN: 0-13-175563-3.
- Mohammadi, M., M. Haghparast, M. Eshghi and K. Navi., 2009. Minimization and optimization of reversible BCD-full adder/Subtractor using genetic algorithm and don't care concept. *Inter. J. Quantum Inf. Proc.*, 7(5): 969-989.
- Nayeem, N.M., A. Md. H. Md, M. Haque, L. Jamal and H.M.H. Babu, 2009., Novel Reversible Division Hardware. *IEEE International Midwest Symposium on Circuits and Systems*, pp: 1134-1138.
- Parhami B., 2000. *Computer Arithmetic: Algorithms and Hardware Designs*. Oxford University Press, New York.
- Parhami, B., 2006. Fault Tolerant Reversible Circuits, *Proceeding of 40th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA.
- Perkowski, M., A. Al-Rabadi, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mishchenko, M.A. Khan, A. Coppola, S. Yanushkevich, V. Shmerko and L. Jozwiak, 2001. A general decomposition for reversible logic. *Proc. RM, Starkville*, pp: 119-138.
- Peres, A., 1985. Reversible logic and quantum computers. *Phy. Rev.*, 32: 3266-3276.
- Toffoli, T., 1980. Reversible computing. *Tech Memo MIT/LCS/TM-151*. MIT Lab for Computer Science.
- Thapliyal, H. and M. Zwolinski, 2006. Reversible Logic to Cryptographic Hardware: A New Paradigm., *49th IEEE International Midwest Symposium on Circuits and Systems*.
- Thapliyal, H., S. Kotiyal and M.B. Srinivas, 2006. Novel BCD Adders and Their Reversible Logic Implementation for IEEE 754r Format, *VLSI Design, Held Jointly with 5th International Conference on Embedded Systems and Design*.
- Vasudevan, D.P., P.K. Lala and J.P. Parkerson, 2004. Online testable reversible logic circuit design using NAND blocks. *Proceeding Symposium on Defect and Fault Tolerance*, pp: 324-331.