

Reversible Signed Division for Computing Systems

D. Krishnaveni, M. Geetha Priya

Abstract—Applications of reversible logic gates in the design of complex integrated circuits provide power optimization. This technique finds a great use in low power CMOS design, optical computing, quantum computing and nanotechnology. This paper proposes a reversible signed division circuit that can divide an n-bit signed dividend with an n-bit signed divisor using non-restoration division logic. The proposed design adequately addresses the ‘delay’ there by improving the efficiency of the circuit. An attempt is made to design a reversible signed division circuit. This paper provides a threshold to build more complex arithmetic systems using reversible logic, thus increasing the performance of computing systems.

Keywords—Low power CMOS, quantum computing, reversible logic gates, shift register, signed division.

I. INTRODUCTION

IN the recent times, a lot of attention is given to the design, implementation, and analysis of logic circuits that are reversible, in research domain. The idea of reversible logic is increasingly employed in the areas of nanotechnology, quantum computing, low power VLSI design, and optical computing. As the complexity in circuit increases, the dissipation of power in the circuit becomes a major challenge in the design. Loss of the information results in dissipation of energy and consumption of power in conventional irreversible logic circuits. It was demonstrated by Landauer [1] that heat energy of $kT \cdot \log_2$ joules is dissipated for every bit of information lost, where the absolute temperature is represented by T (Kelvin) and k is the Boltzmann’s constant respectively. Conversely, it was also shown by Bennet that power dissipated in logic circuits that comprise of logic gates that are reversible is zero [2].

The primary requirement for an ‘n’ input and ‘k’ output logic expression to be reversible is that the values of ‘n’ and ‘k’ must be equal. In reversible logic design, Constant Inputs (CI) are the additional inputs added to fulfill this requirement. The extra outputs added in the circuit to make it reversible are the Garbage Outputs (GO) which do not have any further computational purpose.

Reversible signed division circuit proposed in this paper comprises of Shift registers, parallel adder/subtractors, array of Multiplexers and a control circuit consisting of counter and comparator. To best of our knowledge, reversible gates are yet to be used to design signed division. This design forms the

basis for arithmetic circuits that are reversible. In Section II, the concept of binary signed division is discussed. Section III provides preview of reversible gates used in this paper. In Sections IV and V, the design of proposed Signed Division is discussed followed by simulation results, analysis, and conclusion in Sections VI and VII, respectively.

II. SIGNED DIVISION

For signed binary division, the operands are preprocessed to transform them to positive values.

The dividend is shifted left and the divisor is subtracted from it. If the result of subtraction i.e. the partial remainder is ‘0’ or positive, quotient bit of ‘1’ will be set; the dividend is left shifted and another subtraction is performed. If the partial remainder is negative, a quotient bit of ‘0’ is set and the dividend is restored by adding the divisor to it.

After the normal division process, the results are transformed to the correct signed values, as necessary.

There are two algorithms to perform division operation of binary numbers [3].

A. Restoring Division

Fig 1 shows the logic circuit that performs binary division. At the start of division, Registers M, and Q are initially loaded with n-bit positive divisor, and n-bit positive dividend respectively. Register A is loaded with 0’s. After the completion of division process, registers Q and A contain n-bit quotient and the remainder respectively. The following steps are repeated n times: (i) Shift A and Q left one binary position. (ii) Subtract M from A, and place the answer back in A. 2’s-complement subtraction is performed. (iii) If the sign of A is 1, i.e. if it is negative, set q_0 to 0 and add M back to A (that is, restore A); otherwise, set q_0 to 1.

B. Non-Restoring Division

Non-Restoring Division algorithm avoids the need for restoring partial remainder after an unsuccessful subtraction, i.e. when the subtraction results in a negative result. The following steps are repeated n times for non-restoring division process: (i) Shift A and Q left one binary position. (ii) Subtract M from A, and place the answer back in A. (iii) If the result is positive, compute $2A-M$ by shifting left and then subtracting M. Set q_0 to 1. But if the result is negative, restore it by computing $2A+M$ by adding A and M, shifting left the result and subtracting M. Set q_0 to 0 [3]. Finally, at the end of the process, if the remainder obtained is negative (A), it is restored by adding the divisor (add M to A). After the normal division process, the results are transformed to the correct signed values, as necessary.

D. Krishnaveni is with Department of ECE, Jyothy Institute of Technology, Bangalore, and also a Research Scholar at Visvesvaraya Technological University, Research Resource Centre, Jnana Sangama, Belagavi-590018, Karnataka, India (corresponding author, e-mail: mailkveni@gmail.com).

M. Geetha Priya is with Centre for Incubation, Innovation Research and Consultancy (CIIRC) & Department of ECE, Jyothy Institute of Technology, Bangalore-560082, Karnataka, India (e-mail: geetha.sri82@gmail.com).

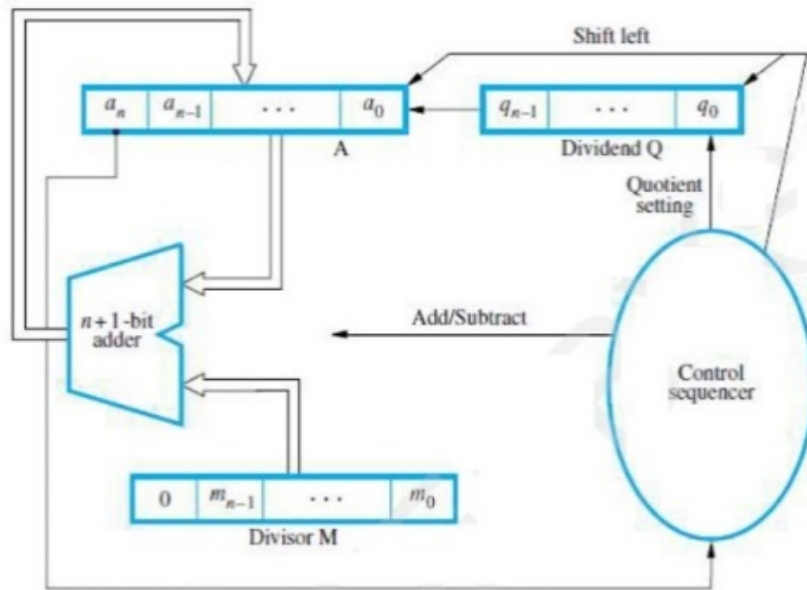


Fig. 1 Block diagram for binary division [3]

III. PREVIEW OF REVERSIBLE LOGIC GATES

A few of the numerous reversible gates that are available in literature are highlighted in Table I along with their quantum cost, input and output vectors.

TABLE I
PREVIEW OF REVERSIBLE GATES

Feynman (FG) [4]: $I_v = (X, Y)$	QC=1
$O_v = (M, N : M = X, N = X \oplus Y)$	
Toffoli (TG) [4]: $I_v = (X, Y, Z)$	QC=5
$O_v = (L, M, N : L = X, M = Y, N = XY \oplus Z)$	
Fredkin (FRG) [4]: $I_v = (X, Y, Z)$	QC=5
$O_v = (L, M, N : L = X, M = \bar{X}Y + XZ, N = XY + \bar{X}Z)$	
Peres (PG) [4]: $I_v = (X, Y, Z)$	QC=4
$O_v = (I, J, K : I = X, J = X \oplus Y, K = XY \oplus Z)$	
SRK [5]: $I_v = (X, Y, Z)$	QC=4
$O_v = (I, J, K : I = X, J = X \oplus Y \oplus Z, K = \bar{X}Y \oplus XZ)$	
DKGP [6]: $I_v = (D, E, F, G)$	QC=15
$O_v = (W, X, Y, Z : W = D \oplus F, X = E, Y = E \oplus F \oplus G, Z = D \oplus (D \oplus F)(D \oplus G) \oplus E(F \oplus G))$	
BVF [7]: $I_v(A, B, C, D)$	QC=2
$O_v = P, Q, R, S : P = A, Q = A \oplus B, R = C, S = C \oplus D$	
SV [8]: $I_v(A, B)$; $O_v = (P = A, Q = \bar{A} \oplus B)$	QC=1

IV. CIRCUIT BUILDING BLOCKS

The proposed design makes use of an array of reversible multiplexers, registers, Universal shift registers, D Flip Flop and parallel adder/subtractors. The design further proposes a control unit that controls the operations of each component in the division circuit and includes a counter and a comparator to check whether operation is completed.

A. Parallel Adder/Subtractor

The 4*4 reversible DKGP gate [6] can work singly as a

reversible Full adder or a Full subtractor. It works as a Full adder or a Full subtractor when the value of input A='0' and A='1' respectively.

A Parallel adder/subtractor is an interconnection of full adders/subtractors to which inputs are simultaneously applied. The carry/borrow generated at a particular stage is propagated to the next stage.

An n-bit reversible parallel adder/subtractor is implemented using the reversible DKGP gates as shown in Fig. 2. When the control input A/S='0', the circuit acts as a parallel adder, thus adding two n-bit binary numbers producing an n-bit sum and a carry out. When the control input A/S='1', the circuit acts as a parallel subtractor, subtracting two n-bit binary numbers producing an n-bit difference and a borrow out.

B. Reversible Multiplexer

The 3 X 3 reversible SRK gates [5] can work as a reversible 2:1 Multiplexer. When input A is 0, information on input line C is selected and placed on output line R. When input A is 1, information on input line B is selected and placed on output line R. Thus, input line A acts as a select line and input lines C and B act as data input lines; forming a 2:1 Multiplexer. The quantum cost of SRK gate is 4.

In the proposed design of n-bit signed division, number of SRK gates (2:1 Multiplexers) used are 'n' (Fig. 3).

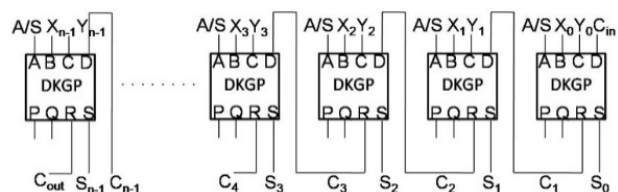


Fig. 2 Reversible Parallel adder/subtractor

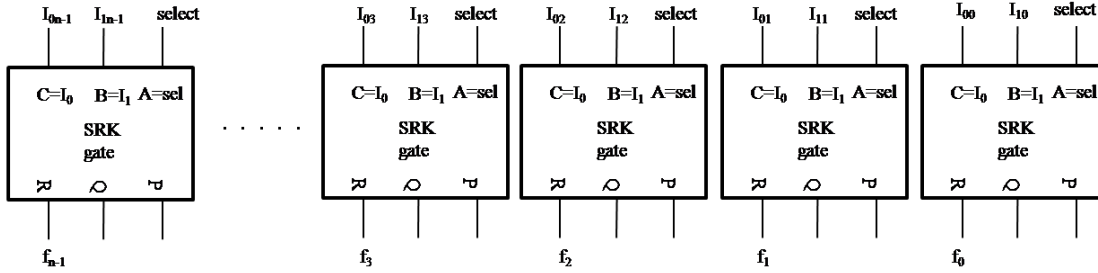


Fig. 3 Reversible SRK gate used as 2:1 Multiplexer (n gates)

C. Sign Flip Flop

A Flip Flop is a bi-stable element that can be used as a one-bit memory device. The characteristic equation of D-Flip Flop (DFF) is $Q^+ = D.clock + Q.clock$. So, a Flip Flop with characteristic equation $Q^+ = D.Lds + Q.Lds$ is implemented using the SRK gate with the input signal Lds acting as clock to the Flip Flop (Fig 4). To avoid a fan out problem, a Feynman gate (FG) is used to copy the output.

When the input signal Lds is high, the sign bit ($d_{n-1}=R_c$) of the dividend is stored in the Flip Flop and when it becomes low, the Flip Flop retains the data.

D. Universal Shift Register

Shift registers are the registers in which the information can be moved position wise upon the occurrence of a clock signal. The information can be shifted in both the directions in a universal shift register. All modes of operation such as Serial in Serial out (SISO), Serial in Parallel out (SIPO), Parallel in Serial out (PISO) and Parallel in Parallel out (PIPO) can also be performed upon the occurrence of clock. Thus, serial data (SIR during right shift and SIL during left shift) or parallel data (Ia, Ib, Ic, Id) can be loaded into shift register. The values at the select lines determine the operation to be performed as given in Table II.

Reversible Universal Shift Register [9] consisting of the Master Slave D Flip Flops (MSDFF) and 4:1 Multiplexer [9] is used in the design of division of signed numbers (Fig 5).

The Master Slave D Flip Flops (MSDFF) [9] has a clocked

D Flip Flop with Asynchronous set/reset input. It makes use of SRK gates, FG gates and BVF gate to act as a D Flip Flop.

Asynchronous set/reset implies that the Flip Flop is set or reset irrespective of the input data and clock. The Flip Flop will set to '1', reset to '0' or operate as a normal D Flip Flop depending on inputs X and Y. When the values of X and Y are '00' or '01', the Flip Flop will reset or set respectively and when XY= '10', it works as a normal D Flip Flop.

Fan out circuits for S1, S0, clock, X, and Y signals make use of FG and BVF gates [9].

TABLE II
FUNCTION TABLE FOR THE REVERSIBLE UNIVERSAL SHIFT REGISTER

S1	S0	Operation
0	0	No Change (Qi)
0	1	Right Shift (Qi-1)
1	0	Left Shift (Qi+1)
1	1	Parallel load (Ii)

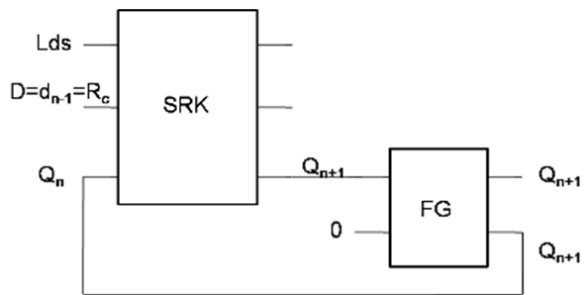


Fig. 4 Sign Flip Flop

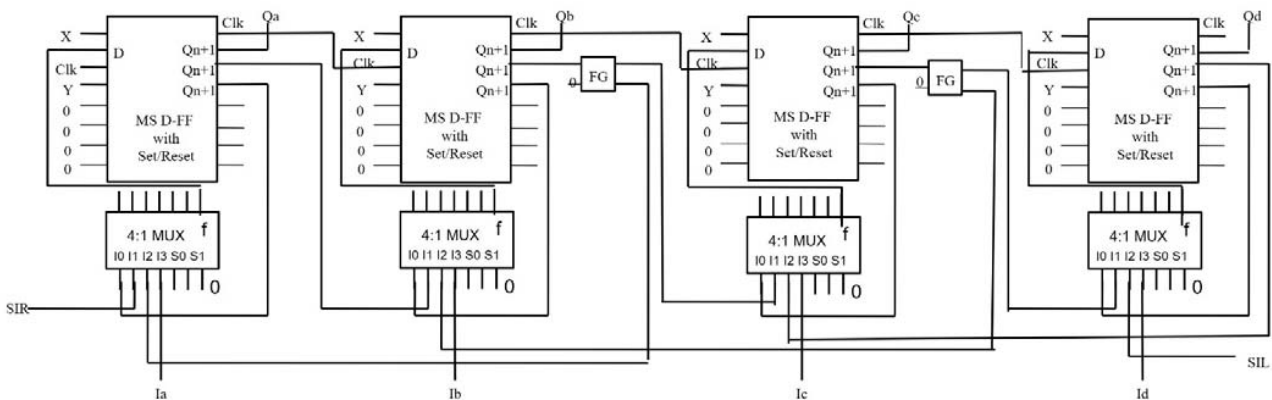


Fig. 5 Universal Shift Register [9]

E. Register

A register is an interconnection of Flip Flops taken as an entity which store information within a digital system so that they can be used later during the computing process. An array of D Flip Flops is used in the design (Fig 6).

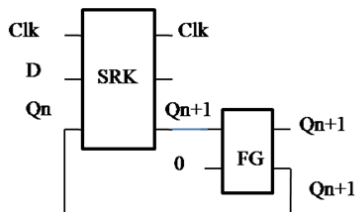


Fig. 6 Simple Reversible D Flip Flop using SRK Gate

F. Control Unit

This unit controls the operation of complete division process by generating control signals and providing them as inputs to various components of the system. The block diagram of control circuit is shown in Fig 7.

Clock signal 'clk', and Enable signal 'E' are given as input to the control circuit. The time period of Enable signal is twice that of the Clock signal. At the start of the division process, the 'Lds' signal is made high only for the first clock cycle. The Select input of the multiplexer array, labelled as 'select' is '0' for first 2 clock cycles and '1' during the remaining operation. The control signals A/S₁, A/S₂, A/S₃, A/S₄ depend on the signals 'k', 'X', 'R_c' and 'R_q'. Select lines S₁ and S₀ to the Universal shift registers are determined from the value of k and Enable signal 'E'. A counter counts the number of shift operations being performed and when the division procedure comes to an end, a signal k='1' is generated which determines the completion of the process. The counter [10] is designed using T Flip-Flops [10]. The T Flip-Flop consists of a PG and an FG gates. Clock given to the counter is given by $clk_{off} = clk.\bar{E}$.

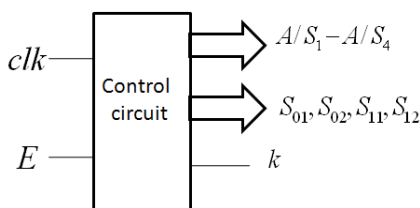


Fig. 7 Block diagram of control circuit

When the number of shifts is equal to the number of bits of the divisor, i.e. after 2n+1 clock cycles, the counter would have counted 2n+1. The output bits of the counter are given as one of the inputs (A-inputs) to an array of SV gates [8]. Reversible 2X2 SV gate works as an EXNOR gate. Second input (B-inputs) to the array of SV gates is the binary value of 2n+1. When both the inputs of the array of SV gates are equal, the outputs X₀X₁X₂...X_{n-1} would be 111...111 i.e. all 1's. These are given as input to an array of PG gates [4] whose output is the signal k. For example, if number of divisor bits is

4, the counter counts 4 ('100') when number of shifts is 4. A-inputs and B-inputs of the array of SV gates are equal to '100' which is the output of the counter. The outputs X₃X₂X₁X₀ of the array of SV gates produce '1111' which is given as inputs 'A₁A₂A₃A₄' to PG gate array [8]. Input-C of each PG gate is '0', which causes output Q to be equal to k='1'.

V. PROPOSED REVERSIBLE SIGNED DIVISION

The proposed design comprises of four Parallel Adder/Subtractors (PAS-I (2n bits), PAS-II (n+1 bits), PAS-III (n bits), PAS-IV (n bits)), two 2:1 Reversible Multiplexer arrays (MUX-I (n+1 bits), MUX-II (n bits)), register to hold n-bit divisor, two left shift registers (here Universal Registers are used- USR-I (n+2 bits), USR-II (n bits)), Flip Flop to store the sign of dividend and a control circuit to control the operation of all devices used in the design.

The design performs division of signed numbers by converting the dividend to a positive number, if it is negative, and storing its sign. PAS-I converts the sign extended dividend to a positive number by finding the 2's complement of the number if it was negative, else keeps it unchanged. USR-I and USR-II are loaded through MUX-I and MUX-II respectively when positive edge of clock pulse occurs and S₁₁S₀₁= S₁₂S₀₂'11'. Initially, when select line 'select' to MUX-I and MUX-II is '0', the converted Sign extended dividend is loaded through the MUX-I and MUX-II to USR-I and USR-II respectively. Most significant bit of result obtained from PAS-II is labeled as 'Y'. Later after two clock cycles, when select='1', the result from PAS-II and quotient bit from USR-II output along with Q₀ = \bar{Y} is chosen by the MUX-I and MUX-II and loaded into USR-I and USR-II respectively when clock is pulsed and S₁₁S₀₁= S₁₂S₀₂'11'. Most significant bit of USR-I is labeled as 'X' which is initially loaded with '0'. Depending on the select lines S₁₁, S₁₂, S₀₁ and S₀₂, the USRs perform the load operation of parallel data (S₁₁S₀₁= S₁₂S₀₂'11') or combined left shift operation on the existing data in USR-I and USR-II (S₁₁S₀₁= S₁₂S₀₂'10'). The entire operations take place at positive edge of the clock.

$$S_{11} = \bar{k} + kX, S_{01} = \bar{k}E + kX, S_{12} = \bar{k}, S_{02} = \bar{k}E,$$

$$A/S_1 = R_c, A/S_2 = R_q \oplus X, A/S_3 = kR_c,$$

$$A/S_4 = k(R_c \oplus R_q)$$

are the control signals given to various components of the design where k is the signal generated from the control circuit, R_c is the sign bit of dividend stored in the sign Flip Flop, and R_q is the sign of the divisor. If the subtraction of the partial remainder and the divisor results in a negative answer, i.e. Y='1', then Q₀='0' and vice-versa. Depending on the sign of the divisor and the X bit, the control signal A/S₂ is obtained which when equal to '1', performs subtraction of the partial remainder and divisor and when equal to '0', performs addition of the above said data. The control circuit counts the number of shift operations performed on the contents of USR-I and USR-II and when the count is equal to 2n+1, where n is

the size of the divisor, the signal $k='1'$ is generated by the control circuit indicating the completion of division process. At the end of the division operation, USR-I contains the remainder and USR-II contains the quotient. At this instant, if X bit is '0', remainder needs no restoration and $S_{11}S_{01} = S_{12}S_{02} = '00'$, thus keeping the data in the registers constant. The quotient and remainder are obtained at the output of PAS-IV and PAS-III respectively. The most significant bit of the quotient and the remainder indicates whether they are positive

(MS bit='0') or negative (MS bit='1'). If $X='1'$, the remainder is restored during the next clock pulse. Finally, if the sign of dividend and divisor are different, quotient is negative and its 2's complement is found by PAS-IV. If the dividend is negative, remainder is negative and its 2's complement is computed by PAS-III.

The proposed Reversible Signed Division design is shown in Figs. 8 (a) and b. The complete operation of the proposed signed division circuit is depicted in flow chart (Fig. 9).

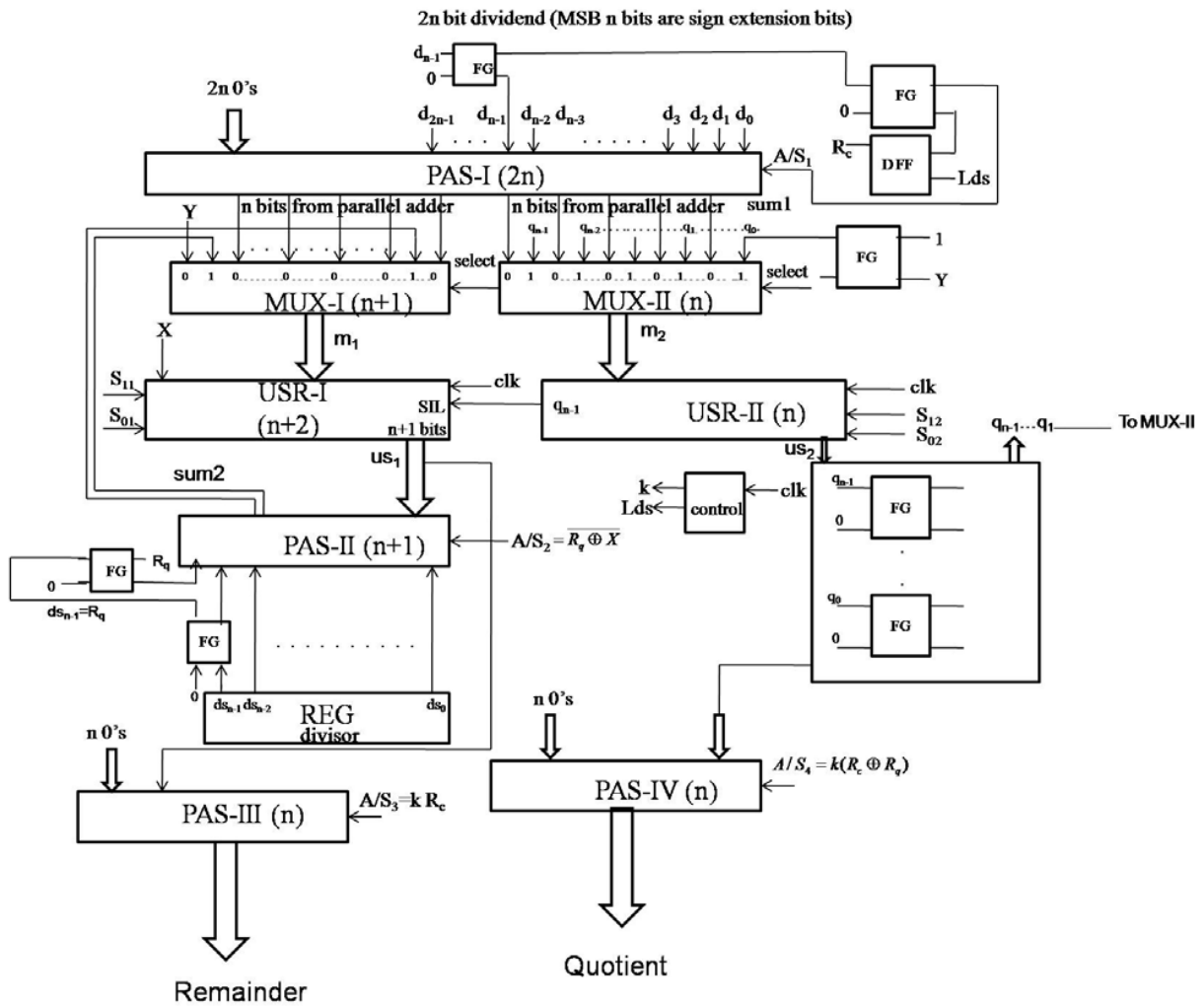


Fig. 8 (a) Design for Reversible Signed Division

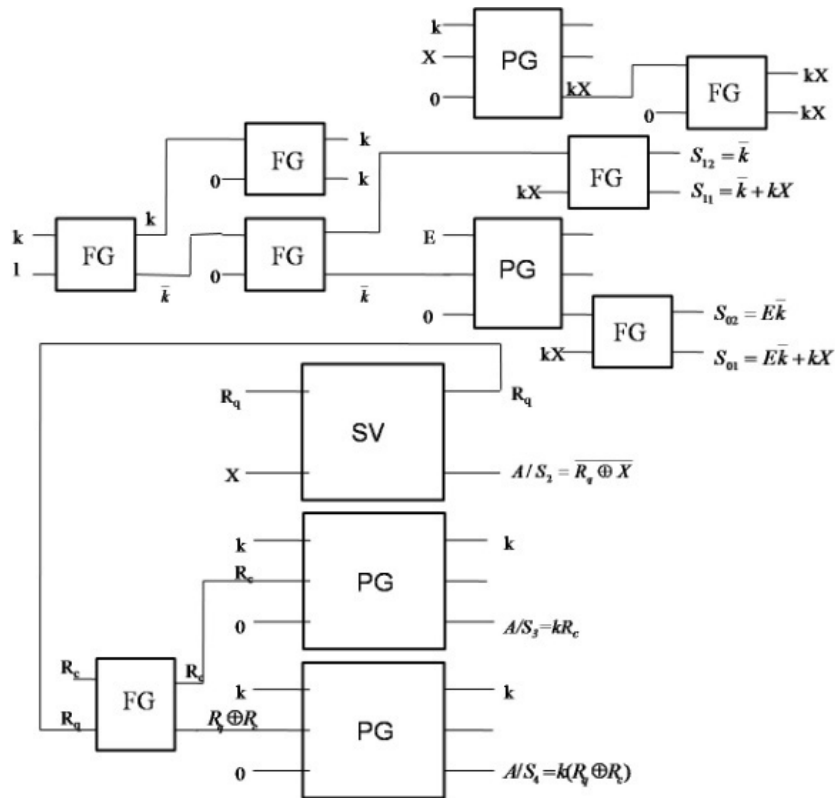


Fig. 8 (b) Generation of control signals A/S2, A/S3, A/S4, and select lines S11, S01, S12 and S02

VI. SIMULATION RESULTS AND ANALYSIS

The dividend and the divisor are taken as 4-bit numbers for simulation. The number of shift operations to be performed for determining the quotient would then be 4. Whether the divisor needs to be added or subtracted depends on the control signal A/S₂ which depends on the sign of the divisor. After 4 shift operations, if the result of addition/ subtraction is negative, the remainder needs to be restored. If the divisor is negative, the remainder is restored by subtracting the divisor, else it is added.

Figs. 10 (a)-(c) show the implementation of signed division algorithm using reversible logic. Fig. 10 (a) illustrates the division of (-7/-2), Fig. 10 (b) illustrates the division of (-7/2) and Fig. 10 (c) illustrates the division of (-13/-3).

Fig. 11 displays the results obtained during simulation of the signed division algorithm using Verilog language and Xilinx and ModelSim software. Figs. 11 (a)-(c) show the results for division of (-13/-3), (-7/-2), and (-7/2) respectively.

Number of gates used is 224 which include the gates used in fan out circuits. CI and GO are 167 and 193 respectively. The Quantum cost of the design (4-bit Reversible signed divider) is calculated to be 808 (Table III).

TABLE III
DIFFERENT PARAMETERS OF PROPOSED SIGNED DIVIDER (4BIT)

	Number of Gates	CI	GO	Quantum cost
Signed Divider	224	167	193	808

Quantum cost of DKGP gate is calculated directly from the quantum circuit. If quantum circuit is reduced using optimization techniques, the Quantum cost of the whole design can be lowered. The Universal shift register used in the design has set reset input. Adequate care has been taken during connections of the gates so that the delay is minimized. In Universal Shift Register, circuits are used to duplicate select lines so that these inputs are given simultaneously to all the components, thus effectively minimizing the delay. There is a tradeoff between delay and the number of gates used. If minimizing the area is the priority, number of gates used would be less at the cost of delay. In this design, priority is given to minimizing the delay. So, large number of gates is used.

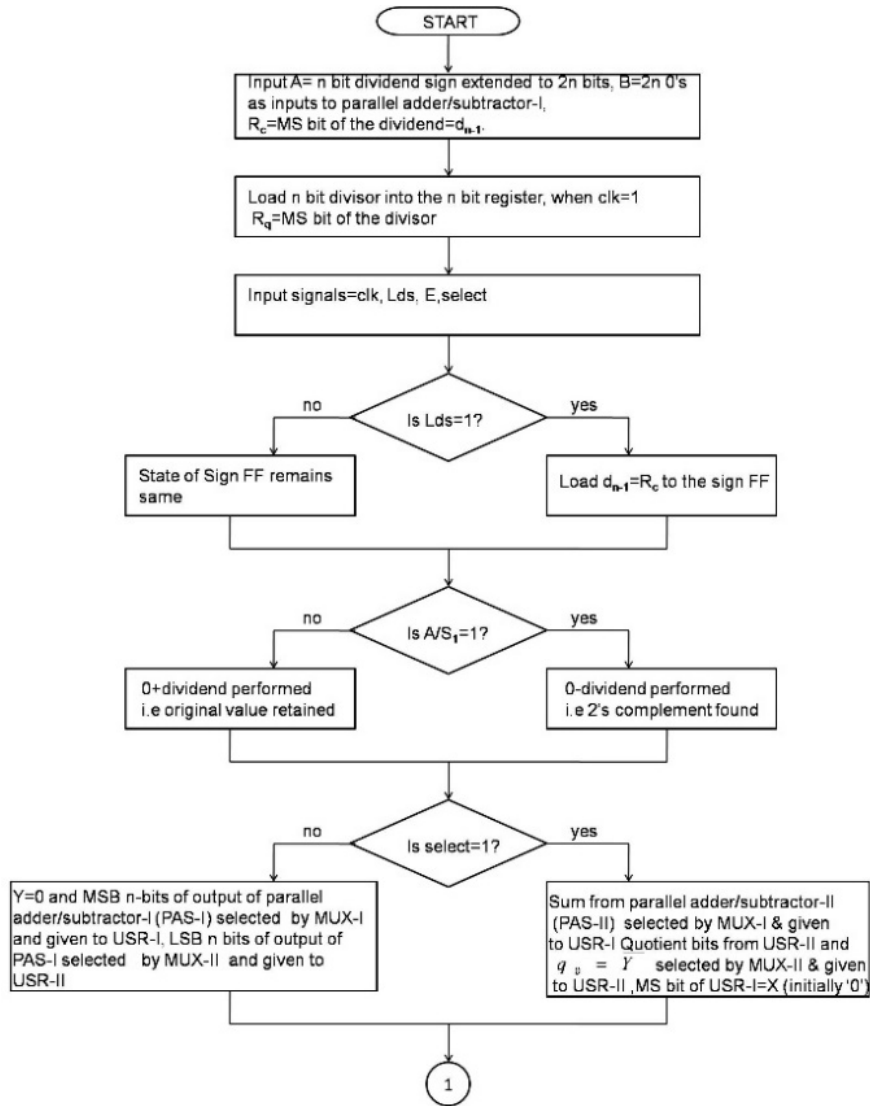


Fig. 9 (a) Flow chart for signed division using reversible logic

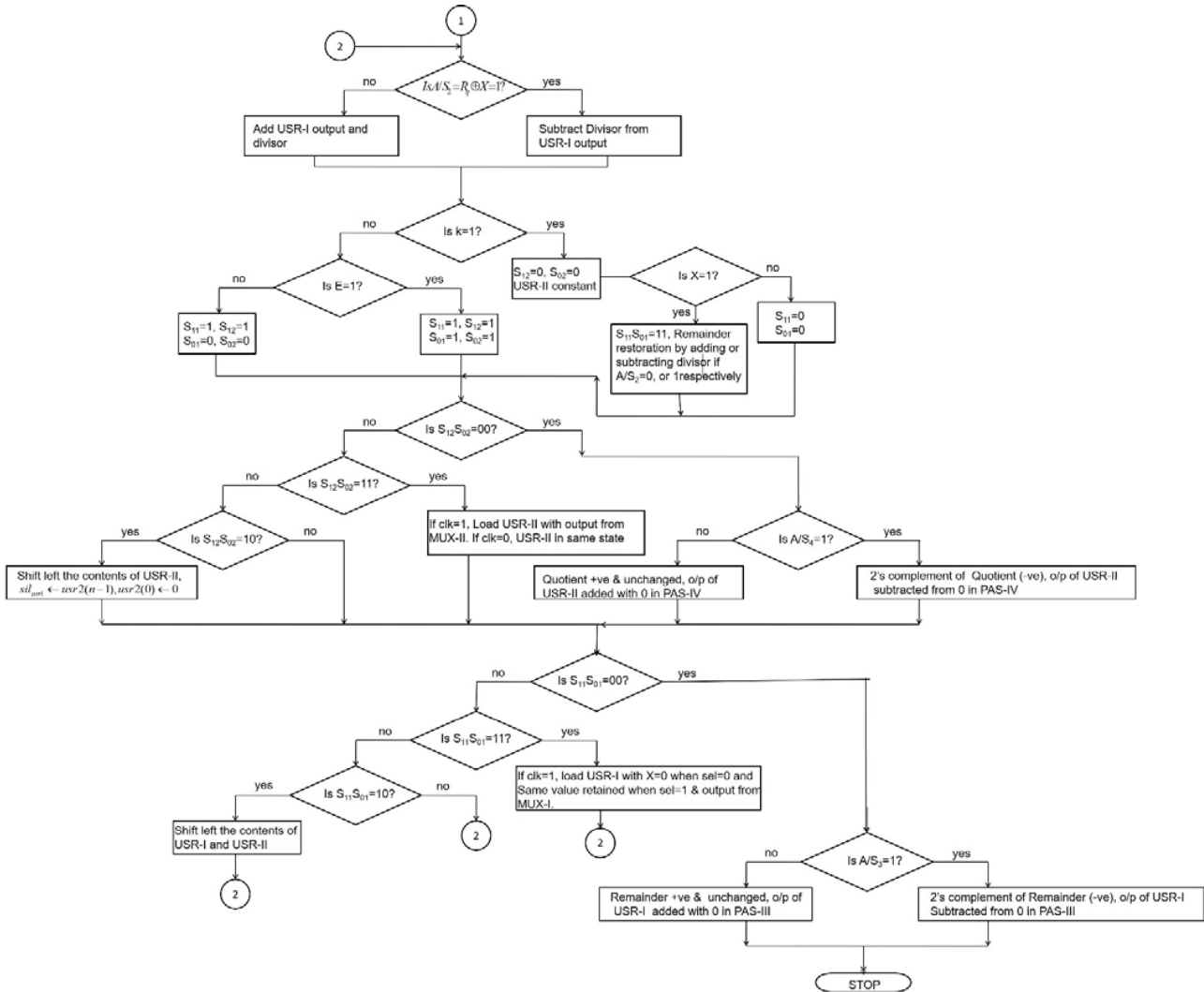


Fig. 9 (b) Contd. Flow chart for signed division using reversible logic

VII. CONCLUSION

The design of Reversible Signed Division is proposed in this paper that takes signed numbers as inputs and generates quotient and a remainder. To the best of our knowledge, reversible gates are yet to be used to design signed division. Thus, this design forms the basis for a reversible arithmetic circuit design and thus contributing to reduced or zero power dissipation.

DIVIDEND=-7=1001, SIGN EXTENDED DIVIDEND=11111001, RC=1, DIVISOR=-2=1110, RQ=1, A3A2A1A0=REMAINDER, Q3Q2Q1Q0=QUOTIENT										
	X	Y	A3	A2	A1	A0	Q3=RC	Q2	Q1	Q0
	X	0	1	1	1	1	1	0	0	1
DIVIDEND -VE, 2'S COMPLEMENT	X	0	0	0	0	0	0	1	1	1
LEFT SHIFT	0	0	0	0	0	0	1	1	1	
ADD			1	1	1	1	0			
	X	1	1	1	1	0	1	1	1	0
LEFT SHIFT	1	1	1	1	0	1	1	1	0	
SUBTRACT			1	1	1	1	0			
	X	1	1	1	1	1	1	1	0	0
LEFT SHIFT	1	1	1	1	1	1	1	0	0	
SUBTRACT			1	1	1	1	0			
	X	0	0	0	0	1	1	0	0	1
LEFT SHIFT	0	0	0	0	1	1	0	0	1	
ADD			1	1	1	1	0			
	0	0	0	0	0	1	0	0	1	1
NO RESTORATION										
RC EQUAL TO RQ, QUOTIENT=0011=3, RC=1, 2'S COMPLEMENT OF REMAINDER=1111=-1										

Fig. 10 (a) Example showing dividend=11111001= -7, divisor=1110=-2 quotient=0011=3, remainder=1111= -1

DIVIDEND=-7=1001, SIGN EXTENDED DIVIDEND=11111001, RC=1, DIVISOR=2=0010, RQ=0, A3A2A1A0=REMAINDER, Q3Q2Q1Q0=QUOTIENT										
	X	Y	A3	A2	A1	A0	Q3=RC	Q2	Q1	Q0
	X	0	1	1	1	1	1	0	0	1
DIVIDEND -VE, 2'S COMPLEMENT	X	0	0	0	0	0	0	1	1	1
LEFT SHIFT	0	0	0	0	0	0	1	1	1	
SUBTRACT			0	0	0	1	0			
	X	1	1	1	1	0	1	1	1	0
LEFT SHIFT	1	1	1	1	0	1	1	1	0	
ADD			0	0	0	1	0			
	X	1	1	1	1	1	1	1	0	0
LEFT SHIFT	1	1	1	1	1	1	1	0	0	
ADD			0	0	0	1	0			
	X	0	0	0	0	1	1	0	0	1
LEFT SHIFT	0	0	0	0	1	1	0	0	1	
SUBTRACT			0	0	0	1	0			
	0	0	0	0	0	1	0	0	1	1
NO RESTORATION										
RC NOT EQUAL TO RQ, 2'S COMPLEMENT OF QUOTIENT=1101=-3, RC=1, 2'S COMPLEMENT OF REMAINDER=1111=-1										

Fig. 10 (b) Example showing dividend=11111001= -7, divisor=0010=2 quotient=1101= -3, remainder=1111= -1

DIVIDEND=-13=10011, SIGN EXTENDED DIVIDEND=-11110011, RC=1, DIVISOR=-3=1101, RQ=1, A3A2A1A0=REMAINDER, Q3Q2Q1Q0=QUOTIENT, Rc=1

	X	Y	A3	A2	A1	A0	Q3	Q2	Q1	Q0
	X	0	1	1	1	1	0	0	1	1
DIVIDEND -VE, 2'S COMPLEMENT	X	0	0	0	0	0	1	1	0	1
LEFT SHIFT	0	0	0	0	0	1	1	0	1	
ADD			1	1	1	0				
	0	1	1	1	1	0	1	0	1	0
LEFT SHIFT	1	1	1	1	0	1	0	1	0	
SUBTRACT			1	1	1	0				
	1	0	0	0	0	0	0	1	0	1
LEFT SHIFT	0	0	0	0	0	0	1	0	1	
ADD			1	1	1	0				
	0	1	1	1	0	1	1	0	1	0
LEFT SHIFT	1	1	1	0	1	1	0	1	0	
SUBTRACT			1	1	1	0				
	1	1	1	1	1	0	0	1	0	0
RESTORATION REQUIRED										
SUBTRACT			1	1	1	0				
			0	0	0	0				
			0	0	0	1				

RC EQUAL TO RQ, QUOTIENT=0100=4, RC=1, 2'S COMPLEMENT OF REMAINDER=-1111=-1

Fig. 10 (c) Example showing dividend=11110011= -13, divisor=1101= -3 quotient=0100=+4, remainder=1111= -1

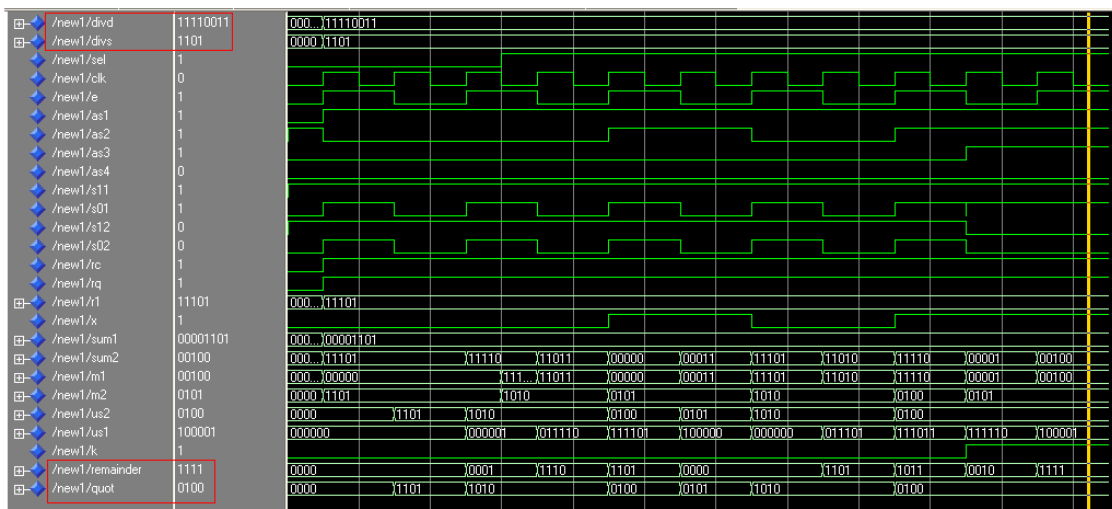
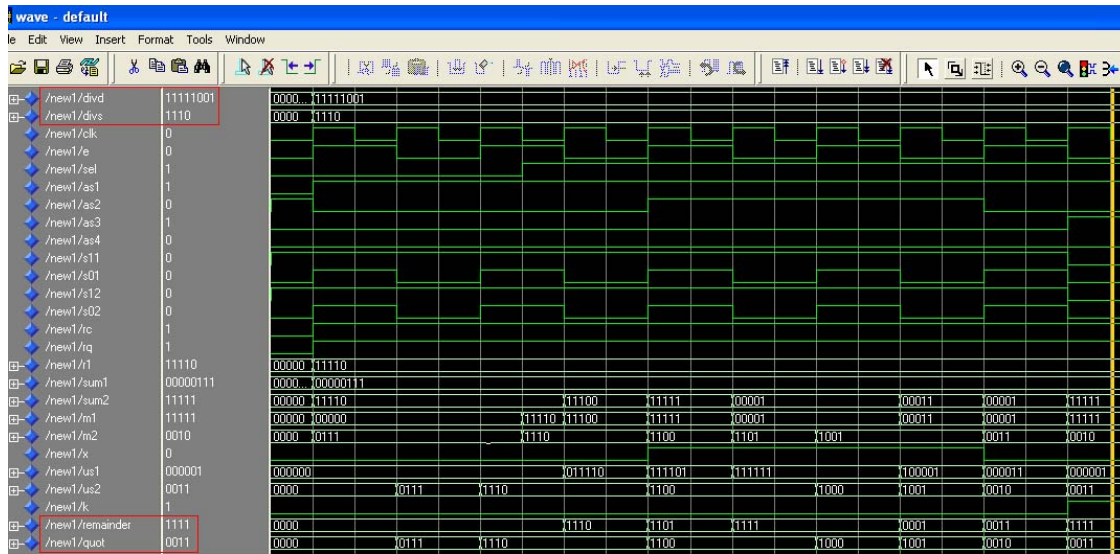
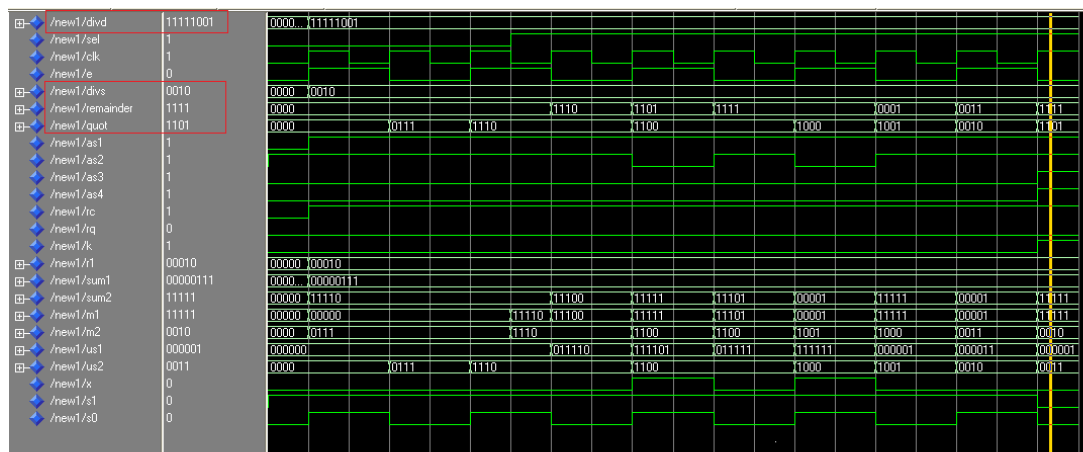


Fig. 11 (a) Simulation of Reversible Signed Division Design with inputs as dividend=11110011= -13, Divisor=1101= -3, Quotient= 0100=+4, Remainder=1111= -1 showing Remainder Restoration



DIVIDEND=-7=11111001, DIVISOR=-2=1110, QUOTIENT=0011=3, REMAINDER=1111=-1

Fig. 11 (b) Simulation of Reversible Signed Division Design with inputs as dividend=11111001= -7, Divisor=1110= -2, Quotient= 0011=+3, Remainder=1111= -1



DIVIDEND=11111001= -7, DIVISOR=0010= 2, REMAINDER=1111= -1, QUOTIENT=1101= -3

Fig. 11 (c) Simulation of Reversible Signed Division Design with inputs as dividend=11111001= -7, Divisor=0010= 2, Quotient= 1101= -3, Remainder=1111= -1

REFERENCES

[1] R. Landauer, "Irreversibility and Heat Generation in the Computing Process", IBM J. Research and Development, vol. 3, pp. 183-191, July 1961.

[2] C. H. Bennett, "Logical Reversibility of Computation", IBM J. Research and Development, pp.525-532, November 1973.

[3] Thapliyal, H. and Ranganathan, N., "Design of Reversible Sequential Circuits optimizing Quantum Cost, Delay, and Garbage Outputs", 2010 ACM J. Emerg. Technol. Comput. Syst. 6, 4, Article 14, (December 2010), 31 pages.

[4] D. Krishnaveni, M. Geetha Priya, 'A Novel Design of Reversible Serial and Parallel Adder/ Subtractor' IJEST vol 3, No 3, March 2011, pg 2280-2288.

[5] D. Krishnaveni, M. Geetha Priya, K. Baskaran, "Design of an Efficient Reversible 8x8 Wallace Tree Multiplier" World Applied Sciences Journal 20 (8): 1159-1165, © IDOSI Publications, 2012

[6] H.R.Bhagyalakshmi and M.K.Venkatesha, 'An improved design of a multiplier using reversible logic gates', International Journal of Engineering Science and Technology Vol. 2(8), 2010, 3838-3845

[7] D. Krishnaveni, M. Geetha Priya, 'A Novel Reversible EX-NOR SV Gate and Its Application' vol 5, LNNS series, Springer, pg 105-114. Net

[8] Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, 3rd Edition, McGraw-Hill, New York, 2002.

[9] D. Krishnaveni, M. Geetha Priya, "A Novel Design of Reversible Universal Shift Register with Reduced Delay and Quantum Cost", Journal of Computing, Volume 4, Issue 2, February 2012

[10] D. Krishnaveni, M. Geetha Priya, "A Novel Reversible n-Bit Counter for Low Power Quantum Computing", International Journal of Control Theory and Applications, Vol.No.10, Issue No.3, 2017, Pg 11-20.