International Journal of Advanced Technology and Engineering Exploration, Vol 8(85)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.874470

Research Article

Designs protracted to combinational and sequential circuits by using hybrid MOS transistor with memristor

V Keerthy Rai and Sakthivel R*
Department of Micro and Nanoelectronics, Vellore Institute of Technology, Vellore, Tamil Nadu, 632014, India

Received: 21-July-2021; Revised: 17-December-2021; Accepted: 20-December-2021

Abstract
A resistive device with a memory characteristic feature entitled memristor is explored to overcome the limitations of the Complementary Metal Oxide Semiconductor (CMOS) technology scaling. The memristor is utilized in place of the Metal Oxide Semiconductor (MOS) transistor due to its non-volatile character and nano-scale element. Thus, researchers worked on a combination of memristor and MOS transistor affords with reduced area exploitation, reduced power dissipation, reliability, and large density. At this juncture, the design of an XOR gate with P-channel Metal Oxide Semiconductor (PMOS) transistors and memristors is accomplished. This design is implemented in 90nm CMOS technology in Cadence Virtuoso and simulations are brought about Spectre. The power dissipation and delay are reduced when compared with conventional CMOS XOR gate. Finally, it is used in full adder design. The average power dissipation is reduced by 69.32%. Further, some of the combinational and sequential circuits like 8×1 Multiplexer (MUX), 1×8 De-multiplexer (DEMUX), and 4-bit Universal Shift Register (USR) are developed. The power of 8×1 MUX is reduced by 43.7 % and the power of 1×8 DEMUX is decreased by 30 % when compared with the conventional designs. The sequential circuit of 4-bit USR is also designed and power is limited by 10.76%. The delay is improved by 79.39% for the proposed 4-bit USR. Improvement of power and delay is observed when compared with the traditional designs.

Keywords
Resistive device, Memory, Memristor, XOR gate, CMOS technology, Combinational circuits, Sequential circuits, 4-bit USR.

1.Introduction
Complementary Metal Oxide Semiconductor (CMOS) technology has struggled with the challenges due to the scaling of the device dimension. Although it is reaching the physical limitations of shrinking the size, it is still expected to be a key aspect of future technology because of the minimum dissipation of static power at higher technology ends. This systematic end of Moore's law finally marks a new era in the research and development of emerging technologies for intelligent computing systems in the future. The researchers build up the interest in memristors which are alternative devices for CMOS transistors and have started working on investigating the memristor for its applications. In addition, memristors have a variety of applications that includes memory computations, neuromorphic architectures, hardware security, speech recognition, and image processing.

The memristor takes its full development during 2008. One of the interests to consider the memristor is a small-scale device of 10nm size for the integration of large density devices [1]. The non-volatility is the vital characteristic feature of the memristor, for the storage of information, computing the logic operations. Some of the applications of memristors are neural networks, analog circuits etc [2].

The functionality of the memristor depends on the Titanium dioxide (TiO_2-TiO_{2-x}) transport mechanism which defines that the oxygen vacancies drift towards its device end. If the positive bias is applied, the oxygen vacancies are moved forward, which increases the impurity width and decreases the resistance of the memristor, called low resistance state R_{ON}. Under the negative bias, the oxygen vacancies are moved backward, which decreases the impurity width and increases the memristor resistance, known to be high resistance state R_{OFF}. As no bias is applied, the impurity width does not

*Author for correspondence

change and exists in the previous memristance state as a memory characteristic [3, 4].

The new way is created to explore computing archetypes by the memristor logic circuits. The memristor-based circuits are the substitute for the traditional CMOS circuits [5]. Because the continuous scaling of the CMOS device accomplishing its scaling restrictions. Hence, conventional Metal Oxide Semiconductor Field Effect Transistor (MOSFET) is replaced by other components like dual halo gate stacked triple material, dual gate Tunnel Field Effect Transistor (TFET) [6], Germanium based dual halo gate stacked triple material surrounding gate TFET [7] for low power applications.

Memristor is also one of the elements for low power applications. Some other reasons to choose memristor circuits are reduction in average power dissipation, the speed of the circuit, and area effective digital design, shrinking reliability [8].

The memristors are used for logic computation is one of the major research interests. The predominant methodology is Material Implication (IMPLY) logic by using memristors. The logic of the IMPLY reveals auspicious results, but more computational steps are needed in the performance logic. The designing circuits of reading/write operation are implemented by using IMPLY logic is completely different when compared with the designs constructed by the Boolean logic. Because of requiring more computational steps, the IMPLY logic is not possible to include in the analysis of Boolean logic. Memristor logic alone has many drawbacks like endurance, reliability, and level restoration [9]. Furthermore, it is not companionable with the current generation of CMOS technology. Further, the logic is developed by hybrid memristor CMOS transistors and the outputs of the hybrid logic have been represented by voltage levels. Such voltage levels are necessary for the CMOS process for signal propagation. These voltages act as a logical state. The basic gates like AND gate and OR gate can be implemented by using memristors only. In this topology, ‘n’ input gates have been constructed using sole memristors, but the implementation of the inverter operation (NOT gate) is not possible by memristors only, instead Metal Oxide Semiconductor (MOS) transistors are also needed. Because the logic family is inappropriate without NOT gate. Also, level restoration has benefited by using hybrid CMOS memristor technology. Digital circuits of any logic computation can be constructed by utilizing hybrid CMOS memristor technology. With the CMOS memristor logic, the estimation of logic gates transient response displays improved performance. The main benefit of the CMOS memristor logic will be the integration of different circuits with CMOS design circuits. Therefore, the input-output signals operate at the level of the same voltages that CMOS logic embraces, thus reducing the necessity of additional circuits. The CMOS logic is defined so far as versatile logic because it proposes several different ways of designing the circuit and architecture according to the necessity of the entire system. Moreover, by using the hybrid memristor CMOS logic, the versatility raises, since the layer of memristor on top of the layer of CMOS, provides the benefit to compute the performances of separate layers and the outcomes are taken from that layer directly [10].

The XOR operation is the main logic function and it is the basic circuit element of different logical entities such as adders, Arithmetic Logic Unit (ALU), binary to the gray code converter, multipliers in computer systems [11]. In this work, the XOR gate using a hybrid CMOS transistor with a memristor is designed. Then full adder is designed using this XOR gate and NAND gate. This gate is compared with the conventional XOR gate. Combinational circuits like binary to gray code converter, Multiplexer (MUX), and De-multiplexer (DEMUX) are also designed using proposed XOR and NAND gates. The binary to gray code converter is employed to rectify erroneous signals in analog to digital converters and digital communication. The MUX plays a vital function in the realization of the control system of signals and circuits for storage. A MUX operation is explained that it selects an input from many input lines and forwarding it to one line of output. Another name of MUX is also known as a data selector [12]. Conversely, the DEMUX function is justified in that it distributes the input from the single line for many lines using selectors. DEMUX is called a data distributor. Finally, the sequential circuit 4-bit Universal Shift Register (USR) is implemented by using the D-Flip Flop (D-FF) and 4-bit MUX. D-FF is designed using a NAND gate. The applications of USR are data transfer, manipulation, and data storage.

Section 2 illustrates the literature review. Section 3 explains the memristor model. Section 4 is represented by the methods of the proposed circuits. The results getting from the simulation and analysis

are addressed in section 5. Section 6 gives a discussion of the work. Finally, the conclusion and future scope have mentioned in section 7.

2.Literature review

The XOR gate was designed with a single memristor and four voltage-controlled MOS transistors. Because of the controlled switches used, it is suffered from area utilization. Over again, it requires many steps to compute [11].

Another XOR design requires three memristors, one CMOS inverter, and a NOR gate with CMOS transistors. For this, the control voltage is required, which is received from the NOR gate. Here also, more usage of the area takes place [13]. One more XOR gate is mentioned, which utilizes four memristors and two transistors. Even though it is area efficient, but it suffers in computation due to the requirement of multiple steps [14]. And another XOR gate is implemented with four memristors, and two transistors based on the crossbar array structure. It is area proficient but it endures with sneak path current due to the crossbar array [3]. By comparing the proposed structure with mentioned structures, it is compatible with area utilization, it requires a single step to perform, it does not expect any control voltages, and it is free from sneak path current by the usage of MOS transistors.

Full adder is considered a crucial element in the architecture of memory computing tasks. It is a basic functional block of the computational elements. The decisive factors of full adders are the consumption of power, area, and transistor count [15]. The conventional CMOS full adder requires 28 transistors. Due to this requirement of more transistors, it suffers high power consumption, propagation delay [16].

The full adder design in the reference paper uses two XOR gates, a pair of AND gates, and a single OR gate. The AND gate and OR gate are designed using a pair of memristors solely. Moreover, the initial resistance of memristors does not affect the computational outcomes. But, these outcomes of the logic could not utilize for the inputs of the next logic directly due to the reason that memristors are not switched devices. The voltage divider condition changes when those outcomes are used as inputs directly as the next logic. To solve this hurdle, CMOS transistors are combined with the memristors. The XOR implementation in the full adder has been done by using six memristors, and an inverter which is implemented by using two MOS transistors [17]. Rather, the full adder in the manuscript is designed using two XOR gates, two NAND gates, one NOR gate, and three inverters. For designing the full adder, more MOS transistors are required in our manuscript when compared with reference paper, but for considering the XOR gate individually, it requires three MOS transistors and three memristors whereas the XOR gate in the reference paper requires six memristors and an inverter with two MOS transistors. Here, the design has the advantage that reduces the number of necessary steps to accumulate the outputs and also eliminates the need for a succession of execution phases and associated driver circuits whereas it faces the complexity of a huge layout area. Thus, the proposed full adder is designed using hybrid CMOS with memristor circuits. The combinational circuit full adder is represented by the Crossbar based-Memristor Ratio Logic (X-MRL) based crossbar array structure. But it has the limitation that suffers the sneak path currents [18].

This problem is overcome by the proposed circuit. In the reference paper, the full adder is designed by using the IMPLY logic requires 21 steps for performing the functionality. If the number of inputs increased, the total number of steps is also increased [19].

The paper in which the full adder has designed by using only memristors. Hence we need additional circuits for signal restoration [20]. The combinational circuits like MUX and priority encoder are also implemented using memristors only. The MUX implementation also requires the computational steps because of using memristors only in this paper [21].

The 4-bit USR is designed by using CMOS transistors, Transmission Gate (TG), and Gate Diffusion Input (GDI) is referred in the papers [22, 23].

The NAND/NOR logic gates are designed based on a memristor that uses the conversion of Voltage to Memristance (VTM) approach. The voltages are inputs and converted to memristance. This memristance is termed as an output. This VTM approach inputs and outputs makes an excellent choice for serving as an interface amid earlier described logic designs to convert one type of logic form to another. The memristance of the output memristor can be taken as logic values in this technique. R_{OFF} and R_{ON} indicate logic 0 and logic 1. The logical state is viewed as data stored in the

output memristor. The power is 3.99 µW and the delay is 202.1ps of the adder using this VTM method. Even though this structure uses only memristors but it has level restoration issues [24].

Another design represents the self-learning capability of universal memristor-based logic. The difficulty of initialization in conventional circuits designed by memristor logic can be overcome by the self-learning modification in the proposed circuits. And this logic family can be easily adaptable to multi-fan-in-logic. The demerit of the self-learning method is the requirement of hardware is large [25].

Additionally, ThrEshold Adaptive Memristor (TEAM) model dynamic full adder circuit, 2-to-1 MUX circuit is introduced. But using dynamic logic, there is a disadvantage of charging and discharging, which results in high power consumption and the circuits are more complex to design. The power determined in this logic is 64.2mW and the delay is 4.49ns [26].

Furthermore, the modified MRL logic gates are implemented which utilizes one transistor and five memristors. Later XOR and AND functionalities are constructed. Soon after full adder, 3-to-8 encoder, 8-to-3 decoder, 4-to-1 MUX, and 4-bit comparators are made. The power of the full adder by using this logic is 6.2mW and the delay is 0.09ns [27]. One more hybrid CMOS memristor full adder is presented. In this design, sum and carry implementations are done individually. So, computational steps are needed in this design [28].

The implementations based on the CMOS transistors have more power dissipation and large area occupation. Consequently, the memristor-based designs need the additional circuits for signal restoration even though the reduced area and power dissipation. In addition, sneak path current issues have been raised when used in crossbar architectures. Moreover, the complete analysis of the literature explains that the proposed designs are implemented using hybrid CMOS transistors and memristors. The bottlenecks raised by the previous designs could be reduced by the proposed designs.

Considering the memristors solely contemplates the challenges. At first, due to the memristor's nonlinear resistance value, the principle and model construction of the circuit is complex, which is detrimental to the development cycle's shortening. Later, the crossbar arrays of the memristor always have the hidden current; accordingly, devices suffer to hold the various logical states. Further, the assumption is that if the researchers control the current of reading/write operation of memristor, the establishment of crosstalk is occurred thereby reducing the parallel output. The first mentioned context depends on the properties of the memristor and couldn't be altered at present. The remaining challenges can be overcome by using hybrid CMOS memristor circuits.

3.Memristor model

The memristor is developed by Chua in 1971 [29]. Further developed in Hewlett Packard (HP) labs in 2008 [30]. The model used in this work is a Voltage ThrEshold Adaptive Model (VTEAM) model. This is developed by Kvantinsky. *Figure 1* shows the memristor model and Voltage to Current (V-I) characteristics. This element gives a relation between charge and flux and is determined by the Equation 1. This Equation 1 represents the pace at which flux changes in relation to the charge.

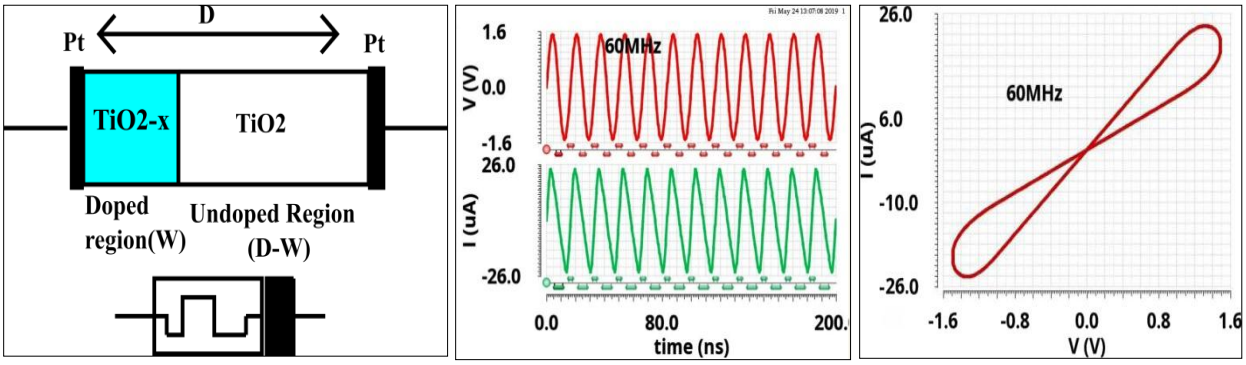


Figure 1 Memristor model, input-output waveforms, and V-I characteristics [30]

$$M = \frac{d\phi}{dq} \quad (1)$$

Where M represents memristance, ϕ is the flux and *q* characterize the charge. And the voltage-current relationship is given by Equation 2. And the VTEAM voltage-current relationship is undefined and can be freely selected from any current-voltage characteristics. Usually, the time-invariant memristive resistance can be controlled by the voltage is presented.

$$v = M(t).i \quad (2)$$

Where the voltage of the memristor is represented by *v*, *i* is denoted as current, *M(t)* is memristance. The detailed memristance M(t) is shown in Equation 3.

$$M(t) = R_{ON}(\frac{w(t)}{D}) + R_{OFF}\left(1 - \frac{w(t)}{D}\right) \quad (3)$$

In the model, the device is divided into two regions. One region is represented by the doped region containing oxygen vacancies (TiO_{2-x}) and it is shown as *w*. Another region is an un-doped region known as the oxide region considered as *D*-*w*. The entire region of the device is committed as *D*. The doped area has more conductance's when compared to the oxide area. When oxygen vacancies are increased by applying the positive bias and become low resistance state R_{ON}, when *w(t)* reached *D* i.e., *w(t)=D*. By applying negative bias, the oxygen vacancies are decreased turned into high resistance state R_{OFF}, when *w(t)=0*. The [*w(t)/D*] is a state variable that varies within the interval (0, D) and it is represented by the internal state *x(t)* described by Equation 6. The derivative of *x(t)* is a function of current and is explained in Equation 4 and Equation 5.

$$\frac{dx(t)}{dt} = f(x, i) \quad (4)$$

Where *x(t)* is an internal state of the memristor.

$$\frac{dx(t)}{dt} = k\, i(t) \quad (5)$$

$$x(t) = \frac{w(t)}{D} \in (0,1) \quad (6)$$

Where the width of the doped area is represented by *w(t)*, length of the memristor is denoted by *D* and *k* provides the relationship between internal state and current. This relationship is characterized by the Equation 7.

$$k = \frac{\mu_v R_{ON}}{D^2} \quad (7)$$

R_{ON} and R_{OFF} are considered as the high and low value of resistance, constant μ_v demonstrates oxygen deficiencies mobility.

4.Methods

The working mechanism block diagram of combinational and sequential logic circuits is shown in *Figure 2*.

The XOR gate is shown in *Figure 3(a)* is proposed using P-channel Metal Oxide Semiconductor (PMOS) transistors and memristors. The operation is explained that when two of the inputs $V_{in1}$and V_{in2} are small, the two PMOS transistors are ON, V_{DD} reverse bias the memristors and the voltage flows through the second transistor and it makes the third transistor OFF. This gives output low. When an input V_{in1} is low and another input V_{in2} is high, the first transistor is in ON state, the first memristor is forward biased and the second memristor is a reverse bias which makes the second transistor is in OFF state such that the third transistor is in ON state. This gives the output high. Conversely, when input V_{in1} is high and another input V_{in2} is low, the first transistor is in OFF state, the first memristor is reverse bias and the second memristor is a forward bias which makes the second transistor is in ON state such that the third transistor is in ON state. This gives the output high. When both inputs $V_{in1}$and V_{in2} are high, the two PMOS transistors are OFF, these input voltages forward bias the memristors and the voltage flows through the memristors and it makes the third transistor OFF. This gives output low. The XOR gate is compared with the outcomes of the traditional CMOS XOR gate [31]. The proposed gate is advantageous when compared with the other gates.

The CMOS gate requires six transistors which have more average power dissipation. The XOR gate is implemented by the memristor with CMOS consisting of six transistors and four memristors which require more area and also have high power consumption when compared with the proposed circuit. The proposed gate consists of three transistors and three memristors which reduces power dissipation and energy consumption.

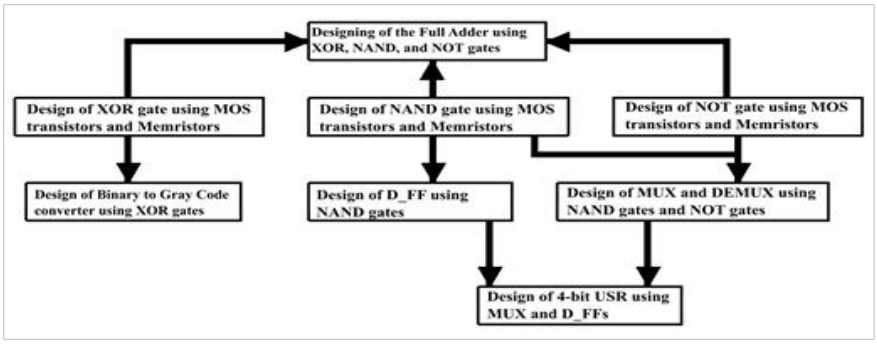
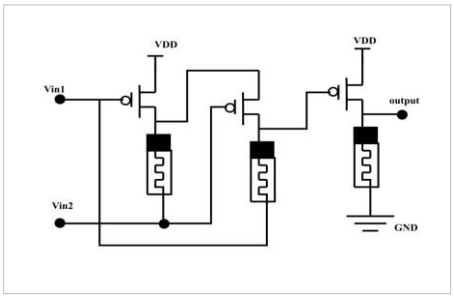
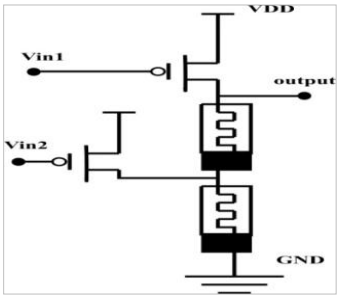


Figure 2 Block diagram of the working mechanism



(a)



(b)

V_{in1}	V_{in2}	Output
0	0	0
0	1	1
1	0	1
1	1	0

(c)

Figure 3 (a) Proposed XOR gate (b) NAND gate (c) Truth table of XOR gate

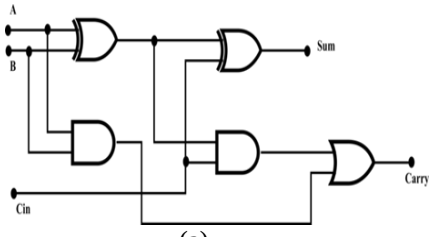
4.1Full Adder using CMOS-memristor

The full adder is designed using the proposed XOR gate and NAND gate with the inverter stage. This NAND gate is shown in *Figure 3(b)* is designed with two PMOS transistors and two memristors and the inverter is designed with one PMOS transistor along with one memristor. Input A and input B are given to the first stage of the XOR gate. Later the first stage output of the XOR gate is given to either input of the second stage XOR gate. Another input of the second stage of the XOR gate is given by input C. Finally, the second stage outcome of the XOR gate gives sum output. Similarly, carry output is obtained by applying the inputs A and B to the first stage of the NAND gate with an inverter. Any of the OR gate inputs is given by this output. The input C is given to one of the inputs of the second stage of the NAND gate with inverter and another input entry of the second stage NAND gate is given by the outcome result of the XOR gate first stage. Then, the output of the inverter with the second stage NAND gate is given to the second input of the OR gate. Finally, the OR gate outcome results carry output. The logical diagram of the full adder is shown in *Figure 4(a)* and is analyzed using Equation 8 and Equation 9. Equation 8 and Equation 9 define the sum and carry functions of the adder circuit.

$$sum = A \oplus B \oplus C_{in} \quad (8)$$

$$C_{out} = (A \oplus B)C_{in} + AB \quad (9)$$

The functional diagram of the full adder using transistors and memristors is shown in *Figure 5*.



(a)

A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b)

Figure 4(a) Logical implementation of full adder, (b) Truth table of full adder

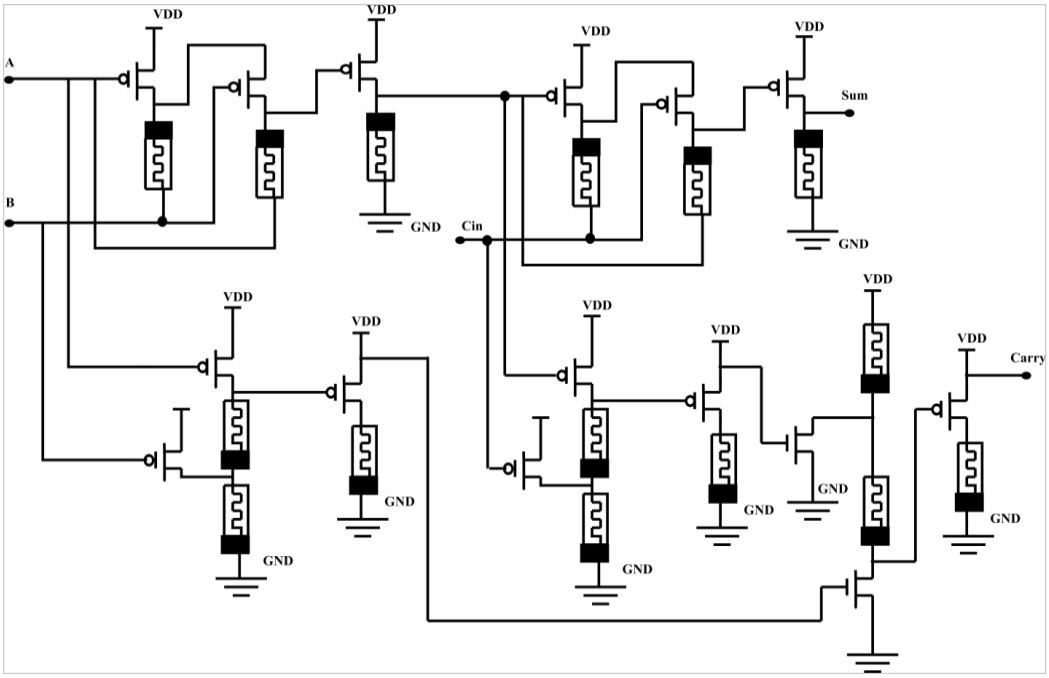
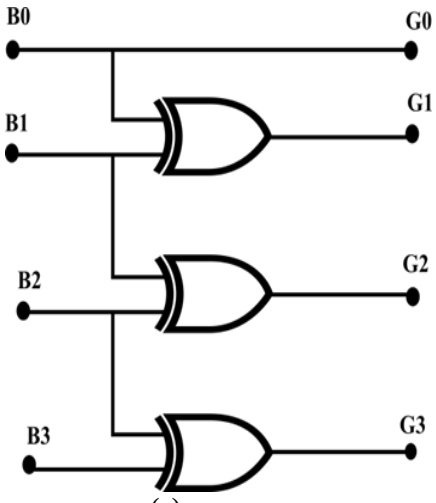


Figure 5 Full adder implementation of the proposed XOR gate and NAND gate

4.2Binary to gray code converter

In computer systems, it is necessary to convert binary to gray code besides gray to binary code. At this instant, the binary to gray code converter is shown in *Figure 6(a)* is designed with the proposed XOR gate and the functional design using transistors and memristors is shown in *Figure 7*. The Most Significant Bit (MSB) of gray code is the same as the MSB of binary code. The remaining bits are getting from the XOR operation of the previous bit and the existing bit.



(a)

B0	B1	B2	B3	G0	G1	G2	G3
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Figure 6 (a) Binary to gray code converter, (b) Truth table

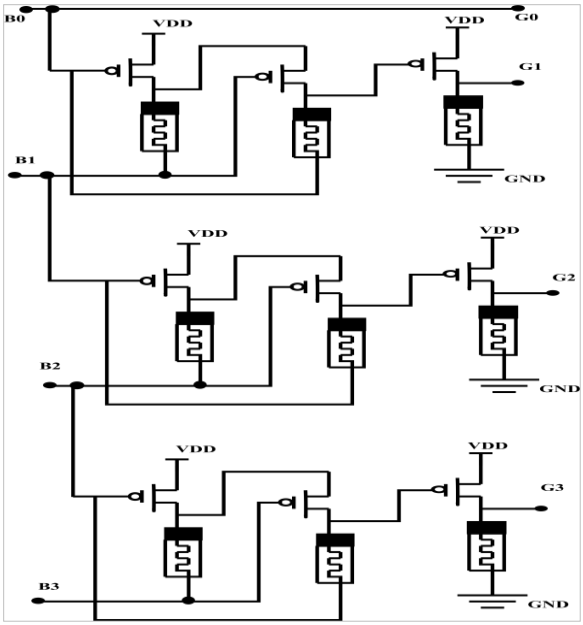


Figure 7 Binary to gray code converter using memristors

The MUX is a circuit with a single output line and many input lines with selection inputs. The operation of the MUX depends on the selection line inputs. The number of input lines depends on the number of selection lines that relate 2^n input lines for 'n' select lines. The logical diagrams of 2×1 MUX, 4×1 MUX are shown in *Figure 8(a), Figure 8(c).* The symbol of 2×1 MUX is shown in *Figure 8(b).*

The proposed 2×1 MUX is shown in *Figure 9* is implemented using the NAND gate designed with a memristor and CMOS transistor with the logic of output Y=S'A+SB [32]. When selection line S is at a low value, the output is taken from input A. On the other hand, selection line S is at a high value, the output receives the input B. After that 4×1 MUX is implemented with three 2×1 MUX. Finally, 8×1 MUX is implemented by using the two 4×1 MUX and one 2×1 MUX. The logical diagram of 8×1 MUX is shown in *Figure 10(a).*

4.3Multiplexer

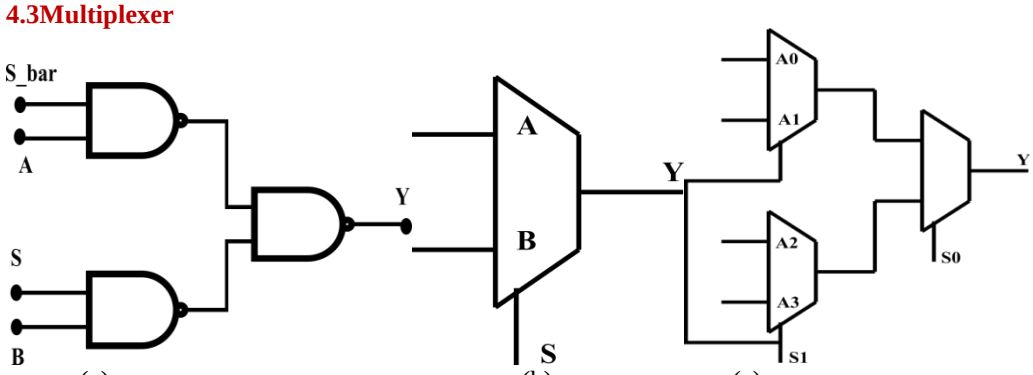


Figure 8 (a)2×1 MUX using NAND gate, (b) Symbol of 2×1 MUX, (c) 4×1 MUX using 2×1 MUX

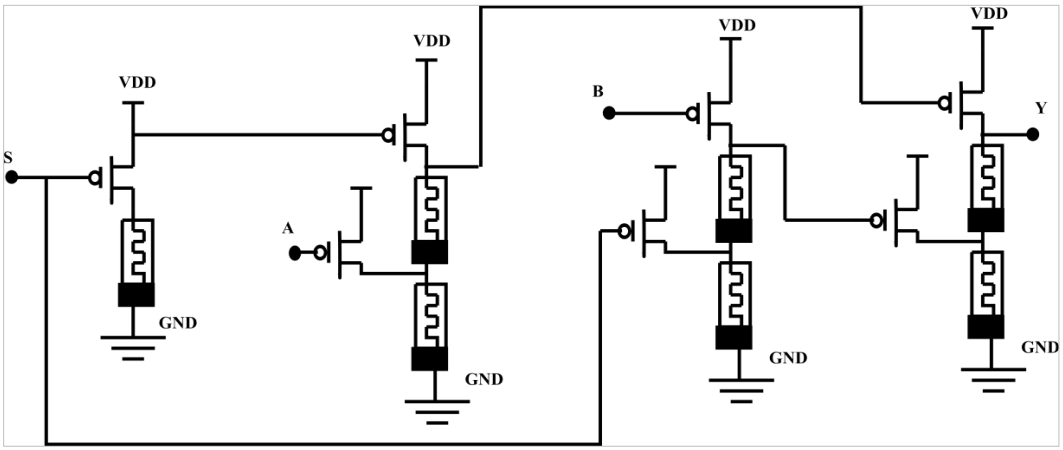
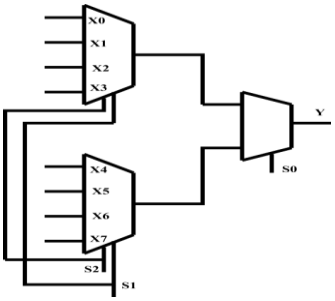


Figure 9 2×1 MUX using memristor



(a)

S0	S1	S2	Y
0	0	0	X0
0	0	1	X1
0	1	0	X2
0	1	1	X3
1	0	0	X4
1	0	1	X5
1	1	0	X6
1	1	1	X7

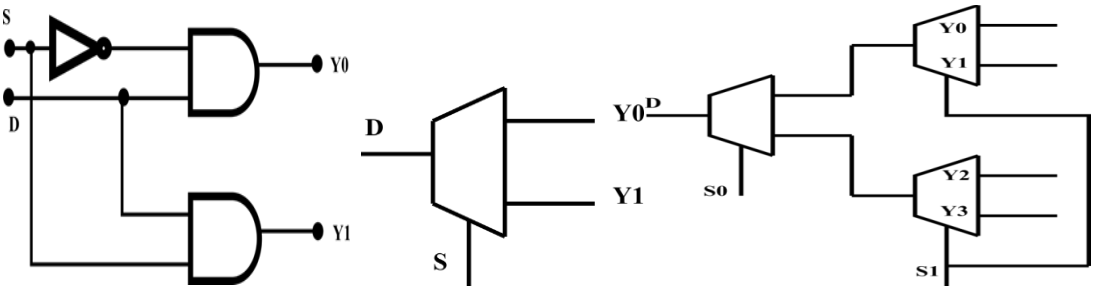
(b)

Figure 10 (a) 8×1 MUX implementation, (b) Truth table of the 8×1 MUX

4.4Demultiplexer

The DEMUX is the reverse operation of the MUX which has a single input line and many output lines. The number of outcomes depends on the select line number. That is 2^n output lines for 'n' select lines. The input is distributed to any of the output lines depending on the select line. *Figure 11(a)* and *Figure 11*(c) show the logical diagrams of the 1×2 DEMUX and 1×4 DEMUX. *Figure 11(b)* shows the symbol of the 1×2 DEMUX.

The proposed 1×2 DEMUX is shown in *Figure 12* is implemented with the logic Y_0=S'D, Y_1=SD. This proposed circuit is designed using memristors and transistors. As the selection line is logic low value '0', the input is distributed to output Y0. Alternatively, the selection line is logic high value '1', the input is disseminated to output Y1. Laterally, 1×4 DEMUX is designed using three 1×2 DEMUX and 1×8 DEMUX is designed using two 1×4 DEMUX with single 1×2 DEMUX. *Figure 13* is shown as the 1×8 DEMUX.



(a) (b) (c)

Figure 11 (a) 1×2 Demultiplexer, (b) Symbol of 1×2 DEMUX, (c) 1×4 DEMUX using 1×2 DEMUX

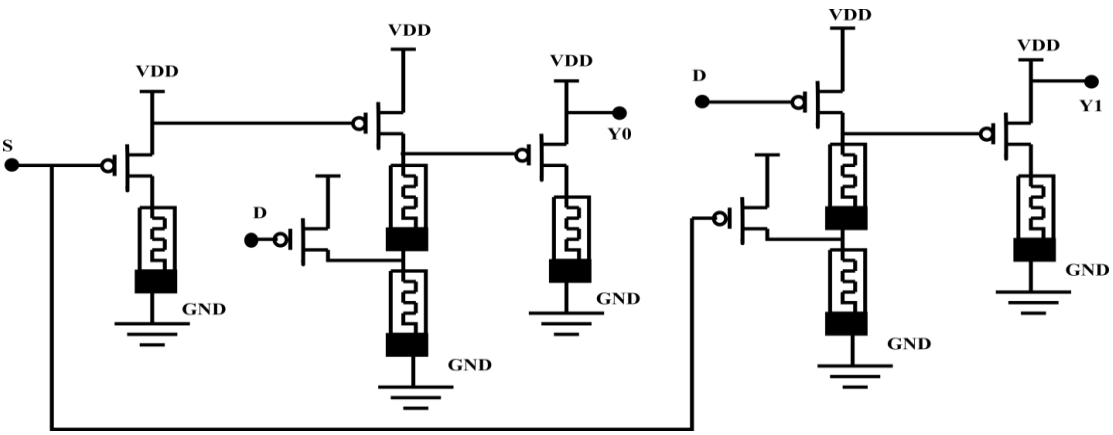
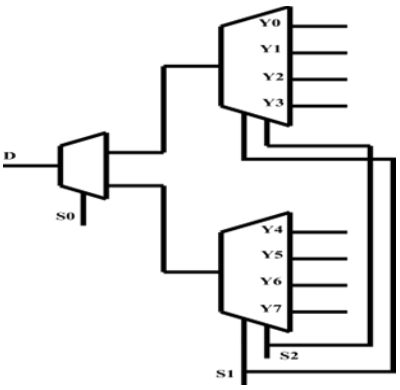


Figure 12 1×2 DEMUX using memristor



S0	S1	S2	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	D	0	0	0	0	0	0	0
0	0	1	0	D	0	0	0	0	0	0
0	1	0	0	0	D	0	0	0	0	0
0	1	1	0	0	0	D	0	0	0	0
1	0	0	0	0	0	0	D	0	0	0
1	0	1	0	0	0	0	0	D	0	0
1	1	0	0	0	0	0	0	0	D	0
1	1	1	0	0	0	0	0	0	0	D

(a) (b)

Figure 13 (a) 1×8 DEMUX (b)Truth table

4.5 4-Bit universal shift register

The registers are used to store information. N-Bit register stores n-bit data. The usage of the register is data storage and processed where ever required. The USR is one sort of register from different varieties of shift registers. A USR has the ability to perform unidirectional shift, bidirectional shift, and parallel loading. It is also served as a memory element [22].
The operating modes of USR are

- Parallel load mode and transmitting the data parallelly.
- Shift operation mode (shifting left and shifting right)
- Hold state.

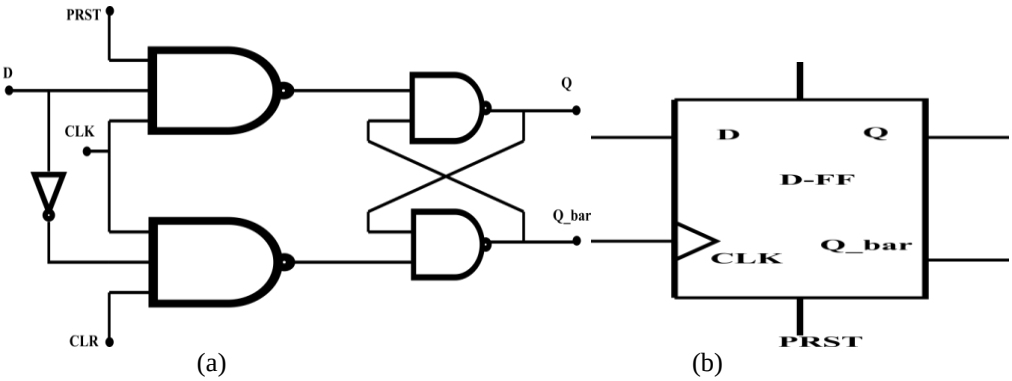
The building blocks of USR are D-FFs and 4×1 MUX. The symbol of D-FF is shown in *Figure 13(a).*

The D-FF is designed using NAND gates which are implemented using CMOS transistors and memristors. The operation of D-FF which is shown in *Figure 14* depends on the clock pulse (CLK), clear (CLR), and Preset (PRST). When CLK=0, PRST=0, and CLR=0, the output Q is holding state. While CLK=1, PRST=1, and CLR=1, the output Q is the same as the input D [33]. The functionality truth table of the D-FF is shown in the *Table 1*.

Table 1 Truth table of D-flip flop

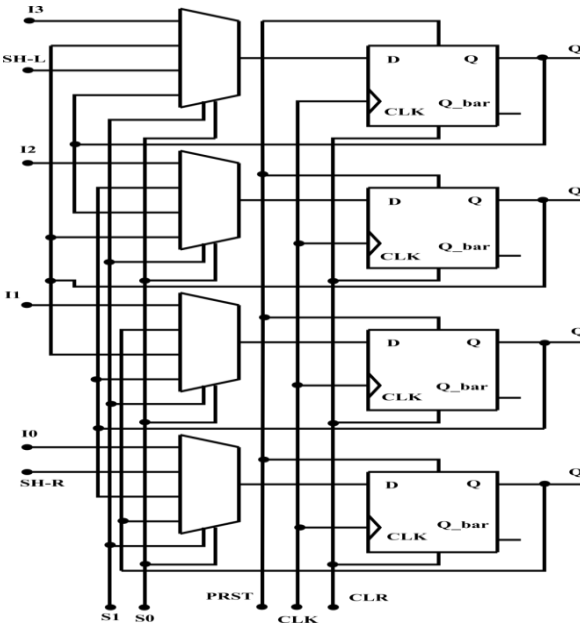
PRST	CLR	CLK	D	Qn+1
0	0	0	x	Qn(Holding state)
1	1	1	0	0
1	1	1	1	1

The 4-bit USR is shown in *Figure 15(a)* has the input SH-R, SH-L as shift right and shifts left. The parallel loads are represented as I_0, I_1, I_2, I_3. PRST, CLR, and CLK control the USR. The selection bits S_0 and S_1 decide the mode of the functionality of 4-bit USR. When S_0 =0, S_1=0, USR operates on parallel load mode. Similarly when S_0 =0, S_1=1, it operates in shift right mode, consequently S_0 =1, S_1=0, USR operates in shift left mode. Finally, S_0 =1, S_1=1, operates in hold condition mode [23].



(a) (b)

Figure 14 (a) D-FF (b) Symbol of D-FF



(a)

S_0	S_1	Mode of functionality
0	0	Parallel load
0	1	Shift right operation
1	0	Shift left operation
1	1	Hold condition

(b)

Figure 15 (a) 4-bit USR, (b) Summarizing of USR functionality is shown in the table

The 4-bit USR requires 4 (21 MOS transistors of MUX) + 4(11 MOS transistors of DFF) = 128 MOS transistors besides 4(21 memristors of MUX) +4 (11 memristors of DFF) = 128 memristors.

The average power is calculated. The total power of design depends on the static and dynamic powers. It is determined by Equation 10.

$$P_T = V_{DD} I_{DS} + C_{PD} V_{DD}^2 f \quad \text{and} \quad I_{lkg} = I_{DS}(e^{\frac{qv}{kT}} - 1) \tag{10}$$

Where V_{DD} is the power supply voltage, C_{PD} is load capacitance and f is operating frequency.
The energy is calculated by integrating the power with respect to time.

The delay is calculated using the following metrics

4.5.1Clock to output (Q) delay

The delay of propagation is formed between the clock input signals to the output signal (Q). It means that the input signal data is set before enough for the input clock signal effective edge.

4.5.2Setup time

The lowest time required amid a change of input signal and the effective edge of triggering input clock signal.

4.5.3Hold time

The lowest time required for the data input to remain stable after the clock input signal triggering edge occurrence.
The propagation delay t_{pd} is calculated by using the Equation 11.

$$t_{pd} \leq T_C - (t_{pcq} + t_{setup}) \tag{11}$$

Where T_c is defined as clock period, t_{pcq} is represented as clock input of sequential element to the propagation delay of output Q, t_{setup} is setup time of the sequential element.

Theoretically, the size of the memristors is small when compared with the MOSFETs, such that the utilization of the area for memristor logic is lesser than the logic of the CMOS. Because of the memristor's width of 3nm which is lesser than MOSFET's width of 90nm. Memristors could be constructed on the MOSFET poly-silicon layer, thus a single MOSFET serves as a stockpile for a large number of memristors. Thus, the transistor count of hybrid CMOS memristor design is reduced when compared with conventional designs and this can be mentioned in analysis tables [17]. Energy consumption is also reduced because of nanoscale memristors due to the reduction of static power leakage. The appearance of glitches is reduced due to the memristor's parameters of the low value of K_{on} and K_{off}.

5.Results

The designs of all circuits were carried out by the Cadence Virtuoso in 90nm technology. 1.8V of the V_{DD} supply voltage used for circuits. The memristor model which is used as the parameters

R_{OFF}=300kohms, R_{ON}=1kohm, k_{ON}=-216.2m/s, k_{OFF}=0.091m/s, V_{ON}=-1.5v, V_{OFF}=0.3v, α_{ON}=3, α_{OFF}=3, x_{ON}=0, x_{OFF}=3nm.

The transient analysis of the different combinational and sequential circuits is evaluated. *Table 2* and *Table 3* give the comparison results of the conventional and proposed full adder designs. The total power is reduced because of the change in the state of equivalent resistance in cascaded memristors for a new combination of the input signals. The consumption of average power value is more for a full adder circuit. This is because of the low R_{ON} and R_{OFF} values of the adopted memristor distinguished to the dynamic resistance of source to drain in MOS transistors, which reduces the leakage current. High values of R_{ON} and R_{OFF} of the memristor could be built up to achieve low power hybrid architectures.

An area of the single memristor is 3nm minimum feature size. As a result, memristors are developed at the top of the CMOS because of their nano-scale and at the level of fabrication compatibility. Consequently, the proposed design simulation shows that the parameter delay time alters the different value input combinations. The conducted simulation shows the time delay value which is affected by the memristor's switching speed, which has controlled by K_{on} and K_{off}. Therefore, the total delay is affected directly by the physical properties of the memristor.

The implementations are made with hybrid CMOS technology, and the memristors are justified. When compared to traditional CMOS technology, constraints such as power consumption, speed, data retention, and endurance are better for hybrid CMOS memristor technology. Eventually, the memristor acts as a switch that depends on the resistance. The resistance of the memristors, as well as variations in the computing environment, determines the power which is required to switch the memristor. But in the proposed design, fewer memristors are used. As a result, switching is minimized which turns to power reduction. A delay is also reduced due to the minimization of the number of computational steps.

The simulation results of the full adder using the proposed XOR gate are shown in *Figure 16*. The parameters of the proposed XOR gate and the full adder which is designed by using the proposed XOR gate are compared with the conventional design. This is shown in *Table 2* and *Table 3*. The power is reduced by 69.32% when compared with the traditional design. The delay of the circuit is also reduced and value 52.56ps.

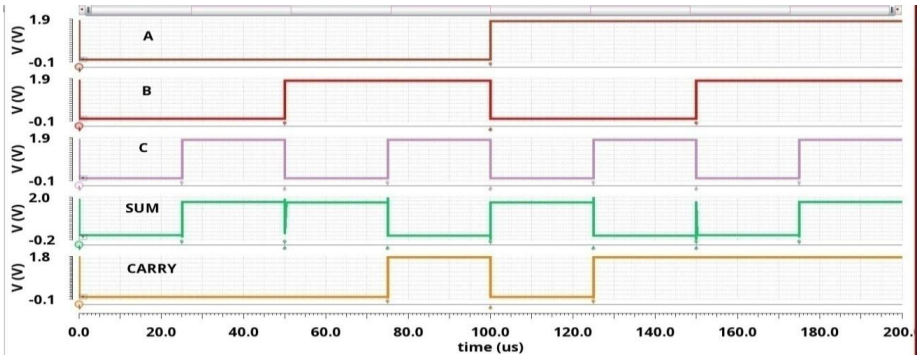


Figure 16 Simulation result of the full adder using proposed XOR gate

Table 2 Comparison of parameters of different designs of XOR gate

Design	Power(W)	Delay(s)	Energy(J)	PDP(J)	EDP(Js)
Conventional XOR gate [31]	722.5n	50.00µ	77.96p	36.13p	3.898f
Proposed XOR Gate	485.4n	50.00µ	73.21p	24.27p	3.661f

Table 3 Performance metrics of full adder design

Full adder design	Transistor count	Power(W)	Delay(s)	Energy (J)	PDP(J)	EDP(Js)
Conventional CMOS [16]	28T	2.4097m	54.94p	234.3844p	0.1324p	12.88z
Proposed Design	15T 15M	739.154µ	52.56p	296.7p	38.85f	15.59z

Binary to gray code converter results are shown in *Figure 17*. The power is reduced by 32.34%. The delay is also reduced and its value is 25ps. The parameters of the binary to gray code converter are shown in *Table 4.*

The simulation results of 8×1 MUX using the proposed 2×1 MUX are shown in *Figure 18*. The power of 2×1 MUX is reduced by 43.25% and the power of 4×1 MUX is reduced by 43%. The power of 8×1 MUX is reduced by 43.7%. The different parameter metrics of MUX are shown in *Table 5*.

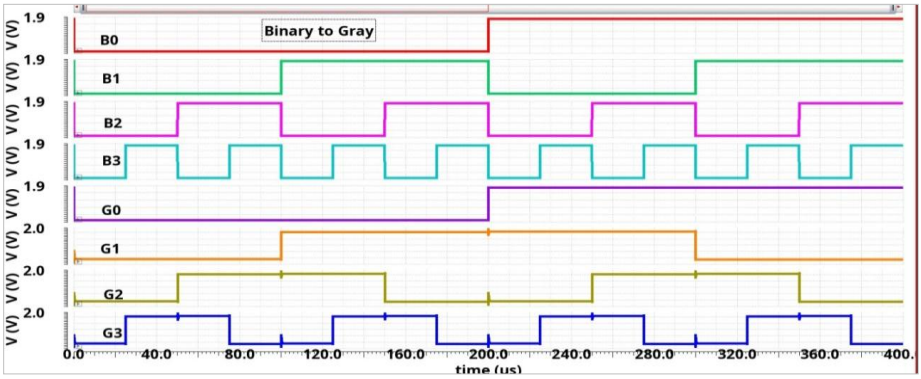


Figure 17 Outcome of the binary to gray code converter simulation

Table 4 Discrimination in parameters of binary to gray code converter

Binary to gray code converter design	Power(W)	Delay(s)	Energy(J)	Transistor count	PDP(J)	EDP(Js)
Conventional CMOS	2.167µ	100.0µ	453.5p	18T	0.2167n	45.35f
Proposed Memristor design	1.466µ	25.00µ	379.3p	9T 9M	36.65p	9.483f

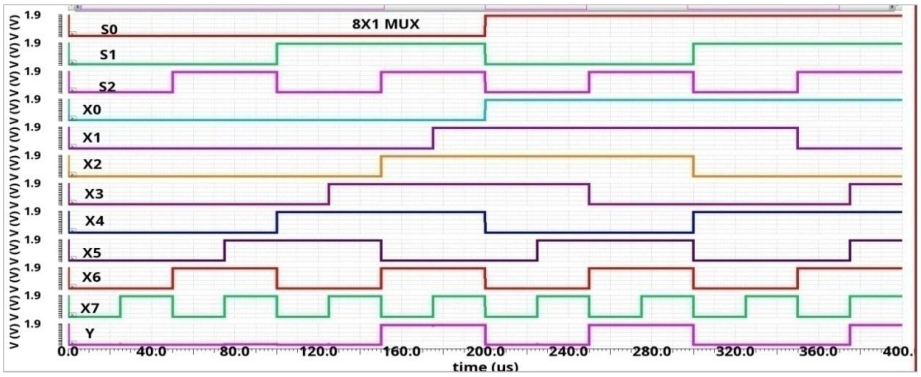


Figure 18 Simulation result of 8×1 MUX using proposed 2×1 MUX

Table 5 Parameter analysis of different MUX designs

Different MUX designs	Power(W)	Delay(s)	Energy(J)	Transistor count
Conventional CMOS based 2×1 MUX design [16]	797.6n	15.34p	67.94p	12T
Proposed Memristor based 2×1 MUX design	452.6n	104.6p	58.44p	7T 7M
Traditional CMOS based 4×1 MUX design	2.393µ	44.90p	526.1p	36T
Proposed 4×1 MUX design using Memritsors	1.362µ	266.8p	322.5p	21T 21M
Traditional CMOS 8×1 MUX design	5.539 µ	76.15p	1.163n	84T
Proposed Memristor 8×1 MUX design	3.118µ	511.9p	703.5p	49T 49M

Experimental results of 1×8 DEMUX using the proposed 1×2 DEMUX are shown in *Figure 19*. The power, delay, and energy parameters are discussed in *Table 6*. The power is reduced by a percentage of 41.15% in 1×2 DEMUX. It is reduced by 35.12% in 1×4 DEMUX. In 1×8 DEMUX, the power is decreased by 30%. The delay is also improved in the proposed design rather than the traditional design and its value is 25µs in 1×8 DEMUX.

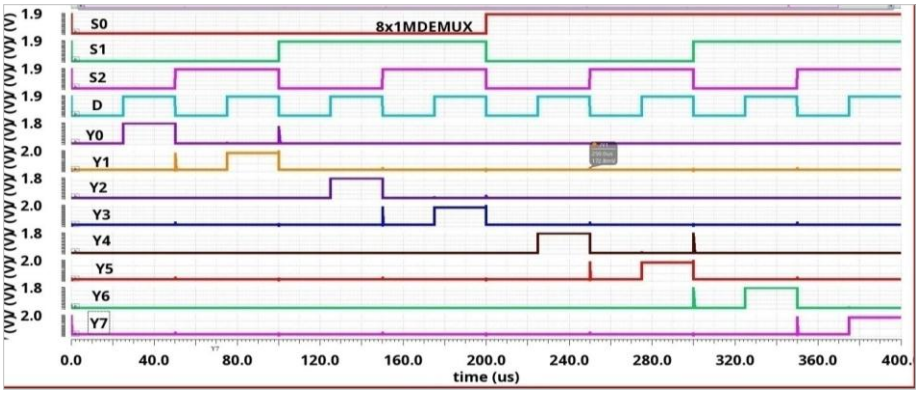


Figure 19 Simulation result of 1×8 DEMUX using proposed 1×2 DEMUX

Table 6 Parameter analysis of different DEMUX designs

Different DEMUX designs	Power(W)	Delay(s)	Energy(J)	Transistor count	PDP(J)	EDP(Js)
Traditional CMOS 1×2 DEMUX	848.3n	100.0 µ	85.01p	14T	84.83p	8.501f
Proposed 1×2 DEMUX using Memristors	499.2n	50.00µ	100.9p	7T 7M	24.96p	5.045f
Conventional CMOS 1×4 DEMUX	2.303 µ	100 .0 µ	224.6p	42T	0.2303n	22.46f
Proposed Memristor 1×4 DEMUX	1.494µ	25.00µ	301.0p	21T 21M	37.35p	7.525f
Traditional CMOS based 1×8 DEMUX	4.970 µ	225.0 µ	1.043n	98T	1.118n	0.2347p
Proposed Memristor 1×8 DEMUX	3.469µ	25.00µ	700.5p	49T 49M	86.73p	17.51f

Figure 20 shows the simulation results of the 4-bit USR using D-FF and 4×1 MUX. It operates when all the control signals PRST, CLR, CLK are high and the operation depends on the selection lines SEL0 (S0) and SEL1 (S1). The constraints like power, delay of the proposed 4-bit USR are compared with conventional design and different designs of USR. Power is reduced by 10.76% when compared with conventional design. The delay is also improved and its value is 2.739ns. This parametric analysis is shown in *Table 7*.

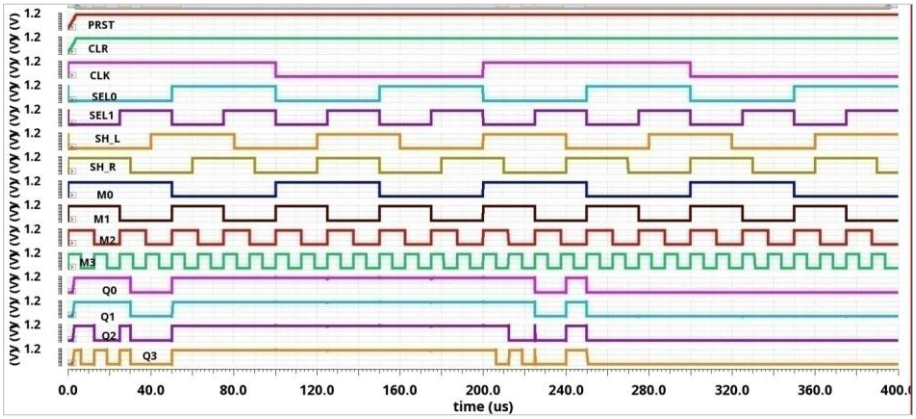


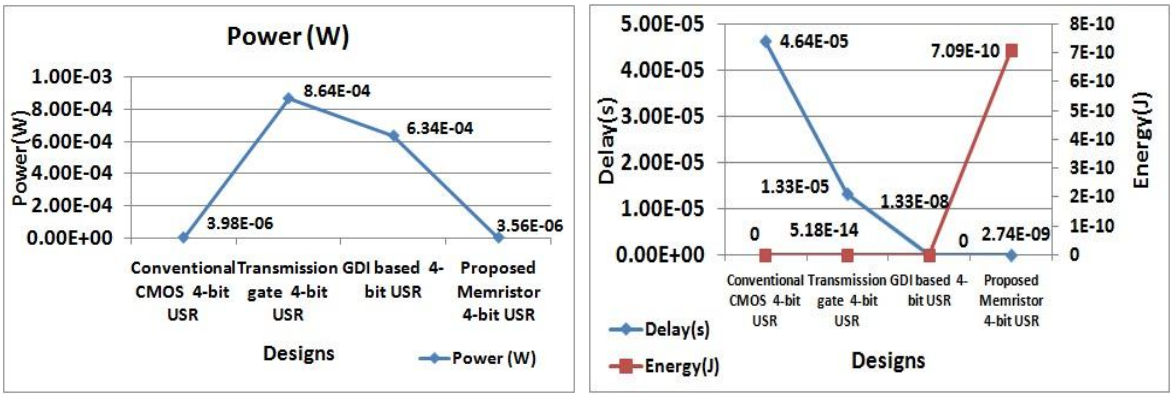
Figure 20 Simulation result of 4-bit USR

Table 7 Power, delay, energy and transistor count of 4-bit USR

Design	Transistor count	PDP(J)	EDP(Js)
Conventional CMOS 4-bit USR [22]	224T	52.95p	0.6878z
TG based 4-bit USR [23]	456T	56.86n	-
GDI based 4-bit USR [23]	312T	29.41n	-
Proposed memristor based 4-bit USR	128T128M	9.737f	1.942a

PDP-Power Delay Product and EDP-Energy Delay Product

The power, delay, and energy plots of proposed design are compared with other designs of 4-bit USR are shown in *Figure 21*(a) and *Figure 21*(b).



(a) (b)

Figure 21 (a) Power of 4-bit USR, (b) Delay and Energy of 4-bit USR

6.Discussion

Because of the nano-scale device memristor, the low power supply reduces power and lowers delay owing to fewer computational steps.

Comparative discussion of the different logics is introduced, the implementation of three-step logic [34], Memristor Aided loGIC (MAGIC) [35], the design of the Memristor as Driver (MAD) logic [36], crossbar based logic structure [37], and resistance-based logic [38]. The circuit construction of the proposed hybrid CMOS memristor logic is fixed compared with other logic designs. Alternatively, different logic operations often correlate to different circuit implementations in the mentioned logic designs might lead to extra fabrication complexity. Because of the unique circuit implementation, crossbar-based logic must cope with the unavoidable sneak path difficulties. After that, the input and output modes represent constant voltages i.e., high level (logic 1), and low level (logic 0). Thus, signal degradation problems arise effectively. Due to the CMOS transistors implemented in the proposed circuit having enough capacity for the load. Later, it is easy to get cascaded the sophisticated logic operations by the acquired output result could be used as a new input signal. Contrast that, MAGIC, MAD logic, crossbar-based designs, and resistance-based logic utilize limiting memristances to represent the inputs and outputs as logic state variables. Because the resistance cannot be obtained or recognized directly in the logic design, hence the four approaches require additional operating circuits. The initialization is not required for the proposed method to operate the basic functions of Boolean logic when distinguished from the other logics. The entire comparison of these logics is summarized in *Table 8.*

Table 8 Comparative discussion of logic implementations

		Input output mode				
Logic implementations	**Circuit construction**	**Input**	**Output**	**Load capacity**	**Easy to cascade**	**Initialization**
Proposed hybrid CMOS memrsitor logic implementation	Fixed	Voltage	Voltage	Sufficient	Yes	Unneeded
Implementation of three step logic	Unfixed	Voltage	Current	Insufficient	No	Needed
MAGIC Design	Unfixed	Memristance	Memristance	Insufficient	No	Needed
Design of the MAD logic	Unfixed	Memristance	Memristance	Insufficient	No	Needed
Crossbar-based logic structure	Unfixed	Memristance	Memristance	Insufficient	No	Needed
Resistance-based logic	Unfixed	Voltage	Memristance	Insufficient	No	Needed

6.1Limitations

The limitations are incorporated in the paper. When a memristor is in the operating phase, it must be ensured that the maximum limit of current does not exceed, because a maximum current could be damaged the device. The logic circuits have the drawback that they must utilize a high amount of resistance, which occupies the more amount of chip area for micron-level chips.

7.Conclusion and future work

The conclusion of this work gives that the proposed XOR gate design with PMOS transistors and memristors gives low power, reduced delay, and low energy consumption when compared with conventional designs. By using the proposed XOR gate, a full adder and binary to gray code converter are designed. Then, the parameters of the proposed full adder and binary to gray code converter are compared with traditional CMOS designs. Further, combinational circuits like MUX and DEMUX are designed. The simulation results show that the average power dissipation is reduced when compared with the standard CMOS design. The delay and energy are also reduced for designs implemented with memristor and MOS transistors.

The implementation of circuits utilized in digital computation is a future possibility of this study. The shift registers can also be used to develop effective storage systems. All abbreviations are listed in *Appendix I.*

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Bürger J, Teuscher C, Perkowski M. Digital logic synthesis for memristors. Reed-Muller. 2013: 31-40.

[2] Liu G, Zheng L, Wang G, Shen Y, Liang Y. A carry lookahead adder based on hybrid CMOS-memristor logic circuit. IEEE Access. 2019; 7:43691-6.

[3] Teimoori M, Ahmadi A, Alirezaee S, Ahmadi M. A novel hybrid CMOS-memristor logic circuit using memristor ratioed logic. In Canadian conference on electrical and computer engineering 2016 (pp. 1-4). IEEE.

[4] Mane P, Talati N, Riswadkar A, Raghu R, Ramesha CK. Stateful-NOR based reconfigurable architecture for logic implementation. Microelectronics Journal. 2015; 46(6):551-62.

[5] Wang HP, Lin CC, Wu CC, Chen YC, Wang CY. On synthesizing memristor-based logic circuits with minimal operational pulses. IEEE Transactions on Very Large Scale Integration Systems. 2018; 26(12):2842-52.

[6] Venkatesh M, Balamurugan NB. Influence of threshold voltage performance analysis on dual halo gate stacked triple material dual gate TFET for ultra low power applications. Silicon. 2021; 13(1):275-87.

[7] Venkatesh M, Balamurugan NB. New subthreshold performance analysis of germanium based dual halo gate stacked triple material surrounding gate tunnel field effect transistor. Superlattices and Microstructures. 2019; 130:485-98.

[8] Singh A. Design and analysis of memristor-based combinational circuits. IETE Journal of Research. 2020; 66(2):182-91.

[9] Reddy KO, Gracin D, Ravi V, Durai SA. Design of combinational logic circuits using memristor and CMOS logic. International Journal of Innovative Technology and Exploring Engineering. 2019; 8(5):167-73.

[10] Singh T. Hybrid memristor-cmos (memos) based logic gates and adder circuits. arXiv preprint arXiv:1506.06735. 2015.

[11] Zhou Y, Li Y, Xu L, Zhong S, Xu R, Miao X. A hybrid memristor-CMOS XOR gate for nonvolatile logic computation. Physica Status Solidi (a). 2016; 213(4):1050-4.

[12] Priyanka MM, Ravi T. An efficient prompt multiplexers using memristor. In international conference on wireless communications, signal processing and networking 2016 (pp. 65-9). IEEE.

[13] Maan AK, James AP. Voltage controlled memristor threshold logic gates. In Asia pacific conference on circuits and systems 2016 (pp. 376-9). IEEE.

[14] Wang X, Yang R, Chen Q, Zeng Z. An improved memristor-CMOS XOR logic gate and a novel full adder. In ninth international conference on advanced computational intelligence 2017 (pp. 7-11). IEEE.

[15] Singh A. Memristor based XNOR for high speed area efficient 1-bit full adder. In international conference on computing, communication and automation 2017 (pp. 1549-53). IEEE.

[16] Yadav AK, Shrivatava BP, Dadoriya AK. Low power high speed 1-bit full adder circuit design at 45nm CMOS technology. In international conference on recent innovations in signal processing and embedded systems 2017 (pp. 427-32). IEEE.

[17] Khalid M, Mukhtar S, Siddique MJ, Ahmed SF. Memristor based full adder circuit for better performance. Transactions on Electrical and Electronic Materials. 2019; 20(5):403-10.

[18] Ali KA, Rizk M, Baghdadi A, Diguet JP, Jomaah J. Hybrid memristor–CMOS implementation of combinational logic based on X-MRL. Electronics. 2021; 10(9):1-17.

[19] Karimi A, Rezai A. Novel design for a memristor-based full adder using a new IMPLY logic approach. Journal of Computational Electronics. 2018; 17(3):1303-14.

[20] Maruf MH, Ashrafi MS, Shihavuddin AS, Ali SI. Design and comparative analysis of memristor-based transistor-less combinational logic circuits. International Journal of Electronics. 2021:1-6.
[21] Karimi A, Rezai A. Novel design for memristor-based n to 1 multiplexer using new IMPLY logic approach. IET Circuits, Devices & Systems. 2019; 13(5):647-55.
[22] Sengupta S, Dastidar A. Evaluation of the performance metrics of a 4 bit universal shift register and its implementation in an arithmetic & logic unit. In international conference on smart technologies for smart nation 2017 (pp. 1576-9). IEEE.
[23] Anandi R. IC layout design of 4-bit universal shift register using electric VLSI design system. International Journal of Engineering Research & Technology. 2017.
[24] Mozafari F, Sharifi MJ, Ahmadi A, Ahmadi M. A novel architecture for memristor-based logic. In international Midwest symposium on circuits and systems 2021 (pp. 812-5). IEEE.
[25] Hong Q, Shi Z, Sun J, Du S. Memristive self-learning logic circuit with application to encoder and decoder. Neural Computing and Applications. 2021; 33(10):4901-13.
[26] Pandey N, Kumar M, Singh S. Realization of memristor based dynamic logic circuits. In emerging trends in industry 4.0 2021 (pp. 1-9). IEEE.
[27] Liu G, Shen S, Jin P, Wang G, Liang Y. Design of memristor-based combinational logic circuits. Circuits, Systems, and Signal Processing. 2021; 40(12):5825-46.
[28] Sharif WMH, Idros MFM, Al-Junid SAM, Osman FN, Razak AHA, Halim AK, et al. Hybrid memristor-CMOS implementation of logic gates design using LTSpice. International Journal of Electrical & Computer Engineering. 2021; 11(3):2003-10.
[29] Chua L. Memristor-the missing circuit element. IEEE Transactions on Circuit Theory. 1971; 18(5):507-19.
[30] Strukov DB, Snider GS, Stewart DR, Williams RS. The missing memristor found. Nature. 2008; 453(7191):80-3.
[31] Ravali K, Vijay NR, Jaggavarapu S, Sakthivel R. Low power XOR gate design and its applications. In fourth international conference on signal processing, communication and networking 2017 (pp. 1-4). IEEE.
[32] Verma A, Akashe S. Low power application for nano scaled memristor based 2: 1 multiplexer. In international conference on communication networks 2015 (pp. 33-6). IEEE.
[33] Zheng J, Zeng Z, Zhu Y. Memristor-based nonvolatile synchronous flip-flop circuits. In seventh international conference on information science and technology 2017 (pp. 504-8). IEEE.
[34] Zhou Y, Li Y, Xu L, Zhong S, Sun H, Miao X. 16 boolean logics in three steps with two anti-serially connected memristors. Applied Physics Letters. 2015; 106(23).
[35] Kvatinsky S, Satat G, Wald N, Friedman EG, Kolodny A, Weiser UC. Memristor-based material implication (IMPLY) logic: design principles and methodologies. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2013; 22(10):2054-66.
[36] Guckert L, Swartzlander EE. MAD gates—memristor logic design using driver circuitry. IEEE Transactions on Circuits and Systems II: Express Briefs. 2016; 64(2):171-5.
[37] Xie L, Du NHA, Taouil M, Hamdioui S, Bertels K. Boolean logic gate exploration for memristor crossbar. In international conference on design and technology of integrated systems in nanoscale era 2016 (pp. 1-6). IEEE.
[38] Zhang Y, Shen Y, Wang X, Cao L. A novel design for memristor-based logic switch and crossbar circuits. IEEE Transactions on Circuits and Systems I: Regular Papers. 2015; 62(5):1402-11.

V Keerthy Rai received the B.E degree in Electronics & Communications Engineering from SarawathiVelu college of Engineering, Sholinghur, Vellore, Tamil Nadu, India in 2007, M.Tech degree in VLSI System Design from Sri Venketeshwara College of Engineering and Technology, Chittoor, Andhra Pradesh, India in 2012. Now, she is pursuing her Ph.D. degree in the School of Electronics Engineering at VIT University, Vellore, India. Her research interests include Low Power VLSI Designs, Digital Designs, Memristor Designs.
Email: raikeerthy@gmail.com

Sakthivel R received the B. Tech degree in Electrical & Electronics Engineering from the University of Madras, Madras, India, in 2000, M. Tech degree in Applied Electronics from Anna University, Madras, India in 2004, and Ph.D. degree in Electronics Engineering from VIT University, Vellore, India in 2014. Currently, he is working as an Associate Professor in the school of Electronics Engineering, VIT University, Vellore, India. His research interests include Low Power VLSI Architectures, Digital & Analog IC Design, VLSI Implementation for Cryptography, Memristor Designs, and Neuromorphic Computing. He is a member of IEEE, IEICE, the Indian Society for Technical Education (ISTE), and the System Society of India (SSI).
Email: rsakthivel@vit.ac.in

Appendix I

S. No.	Abbreviation	Description
1	ALU	Arithmetic Logic Unit
2	CLK	Clock
3	CLR	Clear
4	CMOS	Complementary Metal Oxide Semiconductor
5	DEMUX	De-multiplexer
6	D-FF	D-Flip Flop
7	GDI	Gate Diffusion Input
8	HP Lab	Hewlett Packard Lab
9	IMPLY	Material Implication
10	MAD	Memristor as Driver
11	MAGIC	Memristor Aided loGIC
12	MOS	Metal Oxide Semiconductor
13	MOSFET	Metal Oxide Semiconductor Field Effect Transistor
14	MSB	Most Significant Bit
15	MUX	Multiplexer
16	PMOS	P- channel Metal Oxide Semiconductor
17	PRST	Preset
18	SH-L	Shift Left
19	SH-R	Shift Right
20	TEAM	ThrEshold Adaptive Memristor
21	TFET	Tunnel Field Effect Transistor
22	TG	Transmission Gate
23	TiO_2	Titanium Dioxide
24	USR	Universal Shift Register
25	VTEAM	Voltage ThrEshold Adaptive Memristor
26	VTM	Voltage to Memristance
27	X-MRL	Crossbar Based-Memristor Ratio Logic