

## PERFORMANCE EVALUATION OF COUNTER CIRCUIT FOR REVERSIBLE ALU USING QCA AND VERILOG HDL

TIWARI RAJINDER<sup>1,2,\*</sup>, YESHWANTH G. R.<sup>3</sup>,  
SHARAN PREETA<sup>3</sup>, KUMAR ANIL<sup>1</sup>

<sup>1</sup>Dept. of ECE, ASET, Amity University, Lucknow Campus, Lucknow, U.P., India

<sup>2</sup>Dept. of ECE, Model Institute of Engineering and Technology, Jammu, J&K, India

<sup>3</sup>Dept. of ECE, The Oxford College of Engineering, Bangalore, Karnataka, India

\*Corresponding Author: trajan@rediffmail.com

### Abstract

Quantum-dot cellular automata (QCA) exhibit a new paradigm at Nano scale for possible substitution of conventional CMOS technology. Most of the research works in QCA domain have completely ignored the significance of energy consumption constraint in designing circuits. According to the high rate of dimensions shrinkage in conventional CMOS circuits, serious challenges threaten this technology. Quantum-dot cellular automata are a well-known and possible solution for replacement of CMOS technology. QCA is nothing but semiconductor particle whose range is in Nano meters, hence the name nanotechnology. They have wide range of application due to the ability to absorb light and emit in different wavelength. In this paper we see how to design a synchronous counter circuit using QCA Designer and to find the area occupied by that circuit. The main problem for any circuit is power dissipation and here in this paper, we have reduced the power dissipation by reducing the area occupied by that circuit. Later at last we have compared QCA Designed counter circuit with the Verilog Coded counter in terms of complexity and power dissipation.

Keywords: CMOS, QCA cell, QCA Designer, QCA wiring, Quantum Dots, Synchronous Counter, Verilog HDL.

## 1. Introduction

Quantum-dot Cellular Automata (QCA) is introduced to be an alternative for conventional CMOS technology. On-going shrinkage in CMOS circuits' dimensions has produced some vital issues such as leakage currents and high-power consumption and eventually made them ineffective at nano-scale architectures not only QCA's advent gives a solution for nano-scale designing problems, but also its innovative and simple method for illustration of binary logics has attracted lots of attentions. Unlike, the transistor-based technologies, QCA can operate based on Columbic interaction instead of current. The basic unit in QCA is a cell composed of four dots and two excess electrons which lead to two stable states as the first concept of a digital system. Since the introduction of QCA, many studies have been performed to design various logical circuits based on this technology. For instance; declaring the design rules for robust QCA circuits, implementing full adder cell, design of new nano-scale devices such as five-input majority gate, designs for memory cells, power dissipation analysis and array divider designs have been presented. Many attentions are also paid to practical QCA implementations. A QCA cell can be implemented via four techniques (Metal-island, Molecular, Semiconductor, and Magnetic) as is presented. An accurate operation in QCA circuits could be done in a temperature below 7° Kelvin which is necessary to distinguish between states in a QCA cell. However, many investigations have been made regards to the chance of increasing QCA circuit temperature. These attempts showed that physical characteristics could be changed to implement QCA circuits with ability to work in different temperatures [1].

Based on the literature survey, It has been observed that the registers (i.e., shift registers) as well as the counters are the most commonly used devices for the design and implementation of the sequential circuits. Counters have a wide area of use which two of the most important ones are dividing frequency and counting pulse. Hence, many efforts for designing efficient counters are made by researchers with focusing on the implementation based on the conventional scheme used in digital designs. However, the authors have introduced an innovative QCA compatible design for counters using the inherent capability of QCA loop. In this paper, the author mainly concentrates to design a counter circuit which will help us to reduce the number of cells when compared to the available references given below. Thus, based on the density of the components used by the circuit, area as well as size can be optimised which in turn determines the resultant power dissipated by the circuit. Counter is used for sequential circuits and it can use to trigger any other circuit with some delay, as available in the delay circuit, etc. [2-5].

## 2. QCA Cell

QCA cell is the basic building blocks of the Quantum Cellular Automata. They are nothing but a square shaped cell which have four dots called Quantum Dots or Quantum well. The dots are required to hold the electron in which is nothing but storing binary value in them. They form a capacitor junction between two Quantum Dots which is nothing but junction tunnel. This tunnel will help in forming intercellular barriers so that the electrons inside the well are not allowed to escape from the well when it is exited. QCA cells can be of two types based on the location of the Quantum Dots inside the cell as shown in below Fig. 1 [6].

If the dots are at the corner of the cell, then it has 90-degree phase shift. If the dots are at the centre of each side, then it has 45° phase shift.

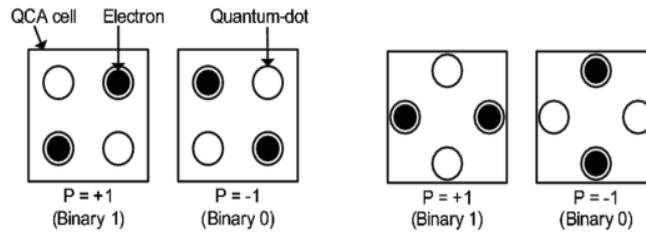


Fig. 1. QCA cell (Arrangement of Quantum Dots inside a Cell) [7].

### 2.1. Polarization

Electrons inside QCA cell will move in diagonal, which is shown in Fig. 1. If the electrons are in bottom-up diagonal, then it has polarization +1 and it carries a binary value 1. If the electrons are arranged in top-down order as shown in Fig. 1, then they have polarization -1 and it carries a binary value 0. The use of polarization is seen in detail when we are going through majority gates [8].

### 2.2. Majority Gates

Majority gates are formed by using four cells around the main cell as shown in Fig. 2. The majority gate is basically 3 inputs 1 output gate where among 4 cells 3 can be considered as inputs and one as output as shown in Fig. 2. Using this, we can create any kind of 3 input basic gates like AND-gate or OR-gate. We can implement two input AND or OR gate by using any 2 out of 3 cells as input cells and one output cell. The remaining one cell will decide whether the given cell is AND gate or OR gate. If the cell is given with polarization -1, then the majority gate is AND gate. If the cell is given with polarization +1, then the majority gate is OR gate as shown in Fig. 2 [9].



Fig. 2. Majority gate with polarization phenomenon [10].

## 3. QCA Wire

In order to carry a given signal, we make use of a wire in traditional analog circuits. Similarly, in this software-based design, we use the term called QCA wire. A QCA wire is nothing but a sequence of cells as shown in Fig. 3.

The working of a QCA wire is simple, i.e., it just carries the electrons in bottom-up or top-down diagonal manner only. So, the value remains the same throughout the wire and the same value is passed out. To keep these cells constant without changing the value, we use the clocking facility in QCA which helps in holding the value inside a cell [12].

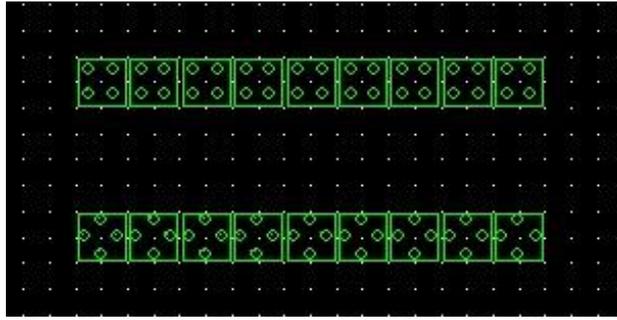


Fig. 3. QCA wire [11].

#### 4.QCA Clocking Sequence

Before going in detail about clocking, we must know how does the cells are differentiated. Fig. 4 shows the colours used to differentiate the cells. The input cell is in blue colour whereas the output cell is yellow. Similarly, the clock cells are green, pink, light blue and white in colour for clock 0, clock 1, clock 2 and clock 3 respectively.

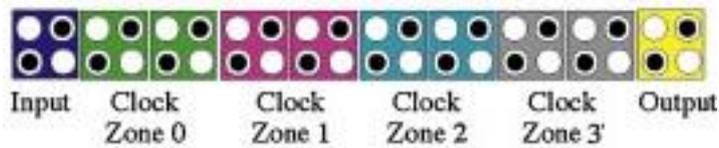
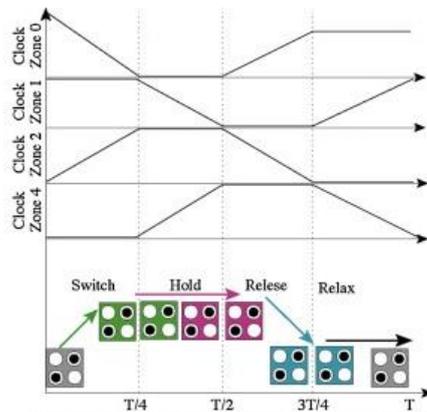


Fig. 4. QCA Clock Sequence [13].

There are four states with respect to four clocks. Each clock represents as state as shown in Fig. 5. Clock 0 represents switch state, where the empty cells will gain electron and get excited from lower level to higher level. Clock 1 represents hold state, where intercellular barriers are formed to hold the electrons. Clock 2 represents release state, where the intercellular barriers are removed and at last clock 3 represents relax state where the electrons are free, and the cell comes back to its normal state. The clocks should be used in cyclic manner to avoid noise in the output [14].



State of Clock Sequencing [14].

### 5. Proposed Design of Counter Circuit

Figure 6 shows the schematic arrangement of the proposed synchronous counter circuit using the QCA tool due to which the design of the circuit becomes quite convenient as well. Performance can be analyzed with working prototypes as depicted in Appendix A. This tool also provides best possible optimization of the power consumption by the circuit.

Figures 7 and 8 show the simulations results of the proposed circuit of the synchronous counter using the QCA tool.

The truth table given below in Table 1 explains the synchronous counter output where, there are four outputs for 4-bit counter and one input called select or sel. If it is a 4-bit synchronous counter then we have  $2^4$  combinations of output, i.e., 16 output combinations as shown above.

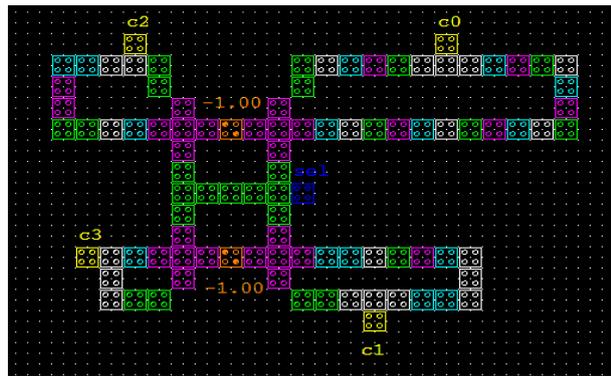


Fig. 6. Proposed design of counter circuit.

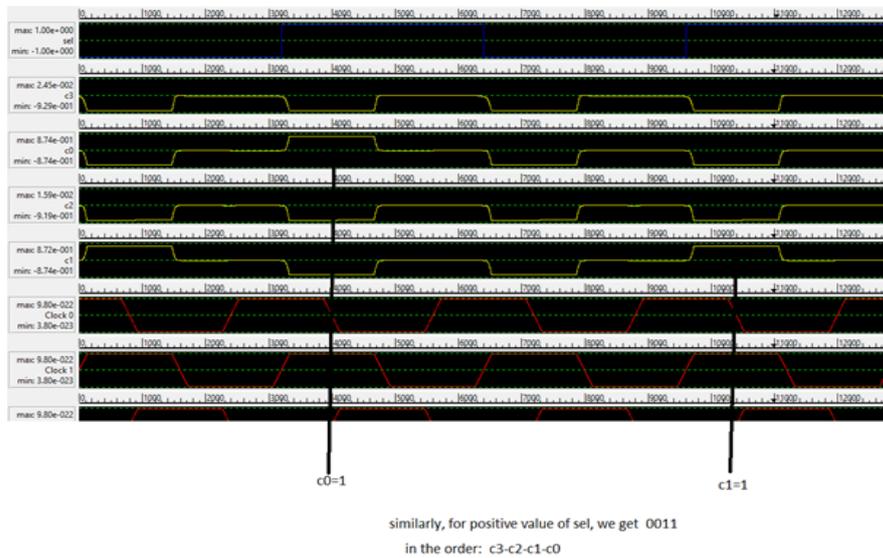


Fig. 7. Simulation result of counter circuit using QCA.



## 6. Performance Evaluation of Counter Circuit on the basis of Area and Power Dissipation

As we discussed earlier, the power dissipated by a circuit can be reduced by minimizing the number of cells and area occupied by the circuit as shown in Tables 2 and 3. Here in this paper, we are comparing our calculations with exiting work [1-4]. Power dissipation can be calculated by the below expression, i.e.,

$$P_{diss} = \frac{E_{diss}}{T_{cc}} < \frac{h}{2T_{cc}} \vec{I} + \left[ -\frac{\vec{r}_+}{|\vec{r}_+|} \tanh\left(\frac{h|\vec{r}_+|}{k_c T}\right) + \frac{\vec{r}_-}{|\vec{r}_-|} \left(\frac{h|\vec{r}_-|}{k_c T}\right) \right] \quad (1)$$

**Table 2. Comparative analysis (Area).**

Reference	Complexity	Area ( $\mu\text{m}^2$ )
[1]	1130	2.20
[2]	652	0.74
[3]	354	0.49
[4]	232	0.20
<b>Proposed</b>	96	0.14

**Table 3. Comparative analysis (Power dissipation).**

Reference	Power Dissipation
[1]	351.22
[2]	395.99
[3]	298.31
[4]	417.83
<b>Proposed</b>	252.14

## 7. Results and Discussion

The Verilog design of the considered circuit will be quite complicated when compared to QCA designed based circuit. In Verilog HDL, the circuit design is nothing but coding which is bit difficult as compared to the circuit design in QCA designer tool as shown in Figures 9 and 10. The steps in simulating Verilog code is more complicated than QCA designer because of the textual coding. Thus, based on the above comparative analysis of the performance of the proposed circuit with the existing one (i.e., references 1 to 4), the author has submitted an innovative and better circuit in terms of complexity, area and power dissipation.

```

20 //////////////////////////////////////////////////
21 module bcdup(
22     input clk,
23     input reset,
24     output reg [3:0] q
25 );
26     always@(posedge clk)
27     begin
28         if(reset)
29             q<=4'b0000;
30     else
31         q<=q+1;
32     end
33     end
34     end
35     end
36     end
37     end
38     end
39     end
40     end

```

**Fig. 9. Source code of main module (Verilog).**

```

24
25 module synbcdutest;
26
27 // Inputs
28 reg reset;
29 reg clk;
30
31 // Outputs
32 wire [3:0] q;
33
34 // Instantiate the Unit Under Test (UUT)
35 asynbcdcu uut (
36   .reset(reset),
37   .clk(clk),
38   .q(q)
39 );
40
41 initial begin
42   clk=1;
43   reset=1;
44   #100:reset=0;
45   #300:reset=1;
46   #100:reset=0;
47 end
48 always
49   #30 clk=~clk;
50
51
52
53
54

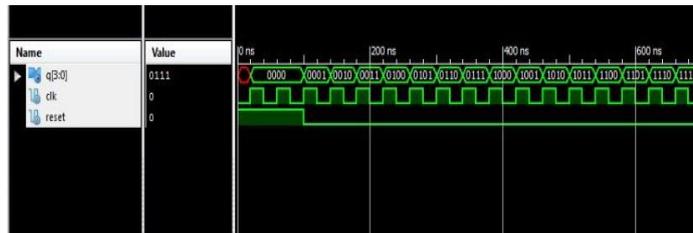
```

**Fig. 10. Verilog code test bench.**

The area occupied by Verilog code is more and power dissipation of the overall Verilog code ends up around 302uW, which is far greater than QCA designer's power dissipation as depicted in Fig. 11.

### Time delay:

One of the main components in Verilog HDL code is giving clock delay, which defines a clock when to rise or fall. So, when we define a delay shown in Fig. 11, the clock sequence will be generated accordingly. Each clock rising edge will have a specified delay so that after the delay the value of the counter will change to the next value.



**Fig. 11. Output waveform (Verilog).**

### Reset:

Another input which acts as enable is reset. Reset is generally used to control the output on or off (which is similar to select input line of QCA Designed circuit). Here initially we keep the reset high so that the value remains in 0000. When the reset is low then the counter starts counting or incrementing for every rising edge clock pulse. In Verilog HDL, as shown in *Appendix B*, the code is divided into 2 main blocks, main module and test bench.

### Main module:

Main module is the first part of writing a Verilog code, where the code for the give logic is written and executed.

Main module gives the complete details of what the code is written for and it can be written in following methods: data flow, behavioural or structural models. In data flow, the code is written in the form of Boolean expressions and in behavioural model, we write the code with the help of truth table and at last in structural model, we write the code in the form of instances.

### Test bench:

In test bench, we have a code as mentioned in Appendix A, which is used to verify the main module. In simple logic, test bench is used to verify the code present in main module. Test bench consist of the values for which the code is verified and delays for each output to appear on the simulator when simulated.

## 8. Conclusions

In this paper we have seen the design of synchronous counter circuit using QCA Designer and Verilog HDL code using XILINX. The area occupied by the Verilog code is more when compared to QCA design. The steps for execution are more in Verilog code is more and complex. If we have error in one step we can't get the output, where as in QCA Designer, the steps of execution are less, and we get output without any interrupts. Clock delay is applied to the Verilog code, in Fig. 10, we can see the there is a delay for clock which is 30 ns.

Also, we can observe that to get the particular output we are giving delays for each input for a particular instant of time. So, this will make a huge impact on overall performance of the code. So, if we see the output of the Verilog code, we can observe there is a delay for each output. On summing the delays, it will cost us more time. Hence, if we see the QCA circuit output, the clock delay is very much less and even can be neglected.

The output of the Verilog code is undefined or do not care until the first positive edge. So, until then the output is considered as a junk values but due to the absence of the external clocking, it will remove such kind of situations. Due to these disadvantages in Verilog circuit, we see that the power dissipated by the code is more when compared to QCA Designed circuit. On calculating, we ended around 302  $\mu$ W, which is much greater than the calculated value of QCA circuit shown in Table 3.

### Acknowledgement

The author is thankful to Hon'ble C-VI (Additional President, RBEF and Chairman AUUP, Lucknow Campus), Maj. Gen. K. K. Ohri (AVSM, Retd.) Pro-VC Amity University, Uttar Pradesh Lucknow, Prof (Dr.) Ankur Gupta, Director, MIET Jammu, Wg. Cdr. Dr. Anil Kumar, Retd. (Guide and Director, ASET), Prof O. P. Singh (HOD, ECE.) and Prof. Preeta Sharan, Professor and Co-Guide, The Oxford College of Engineering, Banglore for their motivation, kind cooperation, and suggestions.

#### Nomenclatures

<i>QCA</i>	Quantum Computational Automata
<i>P<sub>diss</sub></i>	Power Dissipation
<i>VHDL</i>	Verilog Hardware Description Language

## References

1. Yang, X.; Cai, L.; Zhao, X.; and Zhang, N. (2009). Design and simulation of sequential circuits in quantum-dot cellular automata: Falling edge-triggered flip-flop and counter study. *Microelectronics Journal*, 41(1), 56-63.
2. Sheikhfaal, S.; Navi, K.; Angizi, S.; and Navin, A.H. (2015). Designing high speed sequential circuits by quantum-dot cellular automata: Memory cell and counter study. *Quantum Matter*, 4(2), 190-197
3. Angizi, S.; Moaiyeri, M.H.; Farrokhi, S.; Navi, K.; and Bagherzadeh, N. (2015). Designing quantum-dot cellular automata counters with energy consumption analysis. *Microprocessors and Microsystems*, 39(7), 512-520.
4. Aghababa, H.; Yazdinejad, M.H.; Afzali, A.; and Forouzandeh, B. (2008). Simplified quantum-dot cellular automata implementation of counters. *Proceedings of 7<sup>th</sup> International Caribbean Conference on Devices, Circuits and Systems*, 1-4.
5. Sarmadi, S.; Azimi, S.; Sheikhfaal, S.; and Angizi, S. (2015). Designing counter using inherent capability of quantum-dot cellular automata loops. *International Journal of Modern Education and Computer Science*, 7(9), 22-28.
6. Srivastava, S.; Sarkar, S.; and Bhanja, S.a (2009). Estimation of upper bound of power dissipation in QCA circuits. *IEEE Transactions on Nanotechnology*, 8(1), 116-127.
7. Lim, L.A.; Ghazali, A.; Tji Yan, S.C.; and Fat, C.C. (2012). Sequential circuit design using quantum-dot cellular automata (QCA). *Proceedings of IEEE International Conference on Circuits and Systems (ICCS)*, 162-167.
8. Timler, J.; and Lent, C.S. (2002). Power gain and dissipation in quantum-dot cellular automata. *Journal of Applied Physics*, 91(2), 823-831.
9. Bahar, A.N.; and Waheed, S. (2016). Design and implementation of an efficient single layer five input majority voter gate in quantum-dot cellular automata. *SpringerPlus*, 5:636.
10. Goldhaber-Gordon, D.; Montemerlo, M.S.; Love, J.C.; Opiteck, G.J.; and Ellenbogen, J.C. (1997). *Overview of nano-electronic devices*. The MITRE Corporation.
11. Tougaw, P.D.; and Lent, C.S. (1994). Logical devices implemented using quantum cellular automata. *Journal of Applied Physics*, 75(3), 1818-1825.
12. Lent, C.S.; and Tougaw, P.D. (1997). A device architecture for computing with quantum dots. *Proceedings of the IEEE*, 85(4), 541-557.
13. Tiwari, R.; Bastawade, D.; Sharan, P.; and Kumar, A. (2017). Performance analysis of reversible ALU in QCA. *Indian Journal of Science and Technology*, 10(29), 01-05.
14. Tiwari, R.; Kumar, A.; and Sharan, P. (2018). Design and implementation of 4:1 multiplexer for reversible ALU using QCA. *Proceedings of the 2<sup>nd</sup> International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE 2018)* (in press), Ghaziabad, India.

## Appendix A

### Representation and Figures of Design Charts

In this present work, the author has presented a novel design of the circuit of a sequential device, i.e., a counter, which is to be used for the reversible ALU of the quantum processor. The proposed design of this counter has been modelled for the most efficient performance and as per the requirement of the processor using the QCA domain (Viz. Figs. A-1 and A-2. Later on, this circuit has been simulated with dominant parameters with the help of QCA Designer tool as well as Verilog HDL.

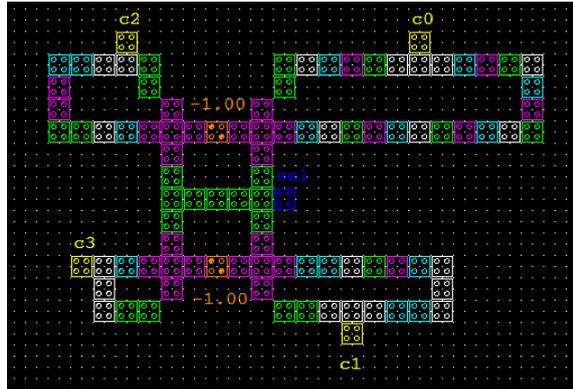


Fig. A-1. Proposed design of the circuit.

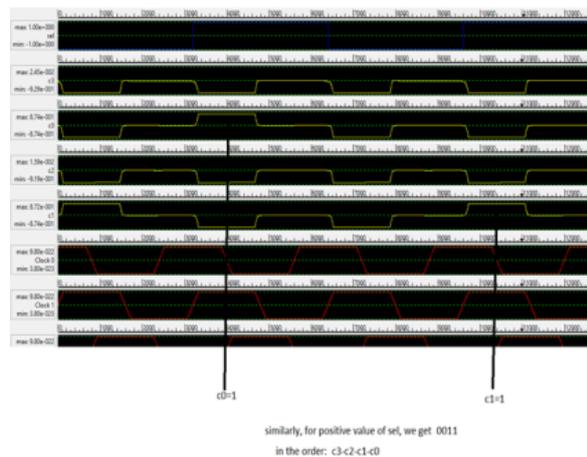


Fig. A-2. Simulated results using QCA designer tool.

## Appendix B

### Verilog HDL Code

#### B.1. Introduction

In the experimental work of this manuscript, the author has made the use of the Quantum Dots Cellular Automata (QCA) Designer tool so as to obtain the best and

desired simulation results. This is nothing but an important tool to simulate a given circuit in nano domain with its desired characteristics and performance.

## **B.2. Programme Structure and Source Code**

In addition to this, some programming has been also carried out using Verilog Hardware Description Language (HDL) in order to obtain the performance evaluation of the proposed circuit of the reversible ALU (Figs. B-1.)

### **B.1. Main Module**

```
module bcdup(input clk, input reset,
output reg [3:0] q);
always@(posedge clk)
begin
if(reset)
q<=4'b0000;
else
q<=q+1;
end
end module
```

### **B.2. Test Bench**

```
module syncbcdutest;
reg reset;
reg clk;
wire [3:0] q;
syncbcdutest uut( .reset(reset), .clk(clk), .q(q) );
initial begin
clk=1;
reset=1;
#100 reset=0;
#300 reset=1;
#100 reset=0;
end
always
#30 clk=~clk;
end module
```

```

20 //////////////////////////////////////////////////
21 module bcdup(
22     input clk,
23     input reset,
24     output reg [3:0] q
25 );
26 always@(posedge clk)
27 begin
28
29     if(reset)
30         q<=4'b0000;
31 else
32     q<=q+1;
33
34
35     end
36
37
38
39 endmodule
40

```

(a) Proposed design of the circuit.

```

24
25 module syncbcdutest;
26
27     // Inputs
28     reg reset;
29     reg clk;
30
31     // Outputs
32     wire [3:0] q;
33
34     // Instantiate the Unit Under Test (UUT)
35     asyncbcd uut (
36         .reset(reset),
37         .clk(clk),
38         .q(q)
39     );
40
41     initial begin
42         clk=1;
43         reset=1;
44         #100;reset=0;
45         #300;reset=1;
46         #100;reset=0;
47     end
48     always
49         #30 clk=~clk;
50
51
52 endmodule
53
54

```

(b) Simulated results.

**Fig. B-1. Verilog code for the proposed circuit.**