

DESIGN AND SIMULATION OF 16 BIT POWER EFFICIENT CARRY SELECT ADDER

S Ranjith Kumar, G Vijaya Bhai Assistant Professor Department Of CSE, ECE Sree Chaitanya College of Engineering, Karimnagar

Abstract: Quantum dot cellular automata, which consume less space and power (quantum cells) at the nanometric scale, are the new technology. The QCA approach seems to be one possible workaround for this physical design, even though CMOS technology does not permit a smaller layout than it does at present time. We constructed the Majority gate using this QCA technology, and the Majority gate computations resulted in the development of a novel Binary coded Decimal (BCD) adder. Verilog programming language and Xilinx ISE software are used to model and synthesize the approach. According to the findings, the suggested approach performs better than the current techniques in terms of area, ADP, PDP, power consumption, and latency (speed). Key terms: CMOS, BCD, and QCA

1. INTRODUCTION

Any VLSI design aims at optimization of any of three parameters namely power, area and delay. Many researchers have achieved this optimization using CMOS technology. CMOS technology gives very promising results and if we try to extend the same CMOS technology to nanometer range the length and width of the channel becomes too small and hence transistor loses its functionality. As alternative CMOS in nanometer scale a new technology QCA (Quantum Cellular Automata) has been developed. QCA is one of the promising technologies that have been employed in modern VLSI design for optimization of power and area. The crucial feature of a QCA cell is that it possesses an electric quadrupole which has two stable orientations. These two orientations are used to represent the two binary digits, "1" and "0". In simplest form QCA is four dot nano cell composed of four dots at corner of the square. The fig.1 given below represents a QCA cell. The four dots of quantum cell represents holes and electrons. White colored dots represent holes and black colored dots represent electrons. Thus, there are four dots in QCA quantum cell out of which two are filled with electrons and two are filled with holes and now if we apply charge, the two electrons are free to occupy any hole and thus we can have two different combinations of holes and electrons in as illustrated in fig.1. These two different combinations of holes and electrons are used to represent the two stable states binary 0 and binary 1 in QCA technology.



Binary 0

Binary 1

Fig.1: Structure of QCA

The QCA are an appealing rising innovation reasonable for the improvement of ultra-thick lowcontrol elite advanced circuits. Hence, over the most recent couple of years, the plan of productive rationale circuits in QCA has gotten a lot of consideration. Exceptional endeavors are coordinated to number-crunching circuits, with the primary intrigue concentrated on the twofold expansion that is the essential activity of any advanced framework. Obviously, the models regularly utilized in customary CMOS plans are viewed as a first reference for the new structure condition. The CFA was an advanced RCA that moderated effects of impending cables. Parallel prefix structures have been dissected and updated in QCA including Bent Kung adder, kogge-stone adder and Han Carlson adder. For the CLA and RCA, increasingly effective structures have been proposed. In this short, an inventive procedure is introduced to execute fast low-zone adders in QCA. Theoretical definitions displayed for CLA and parallel-prefix adders are here mishandled for the affirmation of a novel 2piece extension cut. The last empowers the bring to be multiplied through two following piece positions with the deferral of just a single larger part MG. Similarly, the sharp top level building prompts traditionalist configurations, as needs be avoiding unnecessary clock stages as a result of long interconnections. Aadder arranged as proposed continues running in the RCA style, yet it shows a computational concede lower than all condition of the-workmanship contenders and accomplishes the most decreased ADP.

2. Literature Survey

The 3-input, 1-output majority gate and the NOT gate are the basic gates of QCA technology using which we build universal gate (NAND and NOR) and then any logical circuit can be designed using universal gates. The operation of NOT gate is pretty well known to all where the output is just complement of the input and the operation of 3-input and 1-output gate is a special case where the output is the function of majority of inputs. In otherwords if the two inputs of the 3-input and 1output gate are logic 1 the output is logic 1, conversely if the two inputs of the 3-input and 1-output gate are logic 0 the output is logic 0 and hence, sometimes it is also refereed as 3-input majority voter gate. On similar lines 5Maj gate is the promising one for area and speed optimization and the functioning 5Maj gate is similar to functioning 3Maj gate. The output of 5Maj gate will be logical"1" if at least three inputs of 5Maj gate are logical "1". On same lines the output of 5Maj gate will be logical "0" if at least three inputs of 5Maj gate are logical "0". As a part of area and speed optimization of QCA circuits there were a lot of technological developments had taken place in the last decade and majority of them are listed here. For better understating the developments were reviewed with respect to circuits that are implemented here. Initially we take a glance at technical developments of QCA gates, Adders, RAM"s, BCD adders, and finally ends with clocking schemes. The existing QCAmethod baseddigit serial and parallel decimal adders having the same design, but different from the point of majoritylogics. in [1], [2] literatures, 4-bit ripple carry adder (RCA) based BCD structures has discussed. But, these designs need to be optimized to reduce area requirement. carry look-ahead adder (CLA) approachandCarry flow adder (CFA) approachBCD adders are presented in [3], which showgood delay performance, however it has high hardware complexity. In literature [4] basic binary adder was designed and based on it the proficient4-bit BCD adder was discussed, decreasingwide-ranging consumption. For full usageof the QCA gates, in [5] and [6] correction logic implemented with less majority qca gates. Unlike the offered designs, a new method to calculate carry computation in the multi-bit BCD adder has discussed in this paper. This paperincludedecimal group-generate and decimal group-propagate method to calculate carries initially, and then sums by using the carries in the BCD adder. Thus, This paper have decreasedelay, area, power and hardware complexity in the multidigit BCD adder.

In [7] we can find the basics of Quantum Cellular Automota, the device architecture and the design of 3-input majority gate has laid foundation to the QCA technology and it can also be viewed that it is possible to build quantum devices which can implement logical functions as similar to conventional gates. The implementation of both half adder and full adder using QCA AND-NAND (ANA) and OR-NOR (O-NO) gates was presented in [7]. The adder circuit design is easy to realize practically with minimum no. of gates using QCA technologyespecially with A-NA and O-NO gates. Further other combinational operations like multiplexing, substraction, multiplication, division and other combinational circuits can be effectively designed using the presented adder. The implementation of one bit Quantum full Adder (QFA) and one bit Full subtractor based on their truth table"s is given in[9-10] and gives estimation to build n-bit full adder and n-bit substractor, which

requires 4n and 8n quantum gates respectively. Similarly, implementation of half adder and half subtractor using electron spin is presented based on quantum information processing by neuclear magnetic resonance [11].

3. Proposed Method

The block diagram of proposed 1-digit parallel BCD adder is shown in fig 2. It consists majorly three modules 4 bit binary adder1 unit, Correction logic (CL) using CLA unit and 4 bit binary adder2 unit. As it is a 1 digit BCD adder, it is having the inputs A, B of 4bits applied to Adder1 unit which is shown in fig 3. Now adder1 performs the addition operation between A and B, generates BS(binary sum) and Bcout(binary carry out) as outputs. These outputs are applied to correction logic. It will check the adder1 output is within the BCD boundaries i.e., 0 to 9, If the adder1 output exceeds the value 9, Correction operation will perform and results the correction value as 6. If the adder1 output does not exceeds the value 9, no need haveperformCorrection operation and results the correction logic (CL) using CLA unit.Now, the4 bit binary adder2 is performing the addition operation between the binary adder1 outputs to the Correction logic using CLA output.

The final resultant sum (dS) is the BCD value of A, B and cin addition process.



Fig 2. Proposed 4 bit parallel BCD Adder





The Figure3represents the 1 bit full adder operation, which is implemented by using majority gates and the main use of full adder is we can use it for N-bit addition operations. By using RCA structure, by passing the carry output of first full adder to the next full adder carry input, it is possible to implement N-bit addition. It is also possible to implement By using CLA methodology ,CSKA methodology or PPA methodology. By using this procedure, the 4 bit binary adder1 unit in the Figure1 can be developed.

504



Fig 4. Carry out generation using CL using CLA

The Figure 4 represents the 1 bit Carry out generation process, which is implemented by using majority gates from 4 bits of input. The binary sum, carry out outputs from adder1 unit is applied as inputs here. The majority combinations are used to implement decimal group (dG) and decimal propagate (dP) signals. The dG is signal is generated, if the input bS is greater than or equal to 10. The dP is signal is generated, if the input bS is greater than or equal to 9. Finally by using majority operation between dG, dP and Cin will generate the final resultant carry out. So the final correction value will be generated as CL [3:0] = $\{0, dcout, dcout, o\}$.



Fig 5. Multiple Carry out generations using CL using CLA

The Figure 5 represents the multiple carry out generation process for 4-digit (16 bit) BCD addition process. Finally by using majority operation between dG, dP and Cin(by passing the carry output of first CL using CLA unit to the next CL using CLA unit carry input) will generate the final resultant carry outs dC1,dC2, dC3 and dC4.



Figure5: Proposed 1-bit Full adder for Add2

Figure6 represents the 1 bit full adder operation, which is implemented by using majority gates of 3 input and 5 input. It is possible to use same full adders for both Adder1 unit and Adder2 unit. But in order to reduce the path delays, area and power consumption, the optimized full adder with two majority gates is preferable as shown in above figure. The main use of full adder is we can use it for N-bit addition operations. By using this procedure, the 4 bit binary adder2 unit in the Figure1 can be developed.



Fig 7. Proposed 4 digit (16 bit) parallel BCD Adder

The Figure 7 represents the 16 bit BCD Adder, which performs the BCD addition process parallel manner. All the adder1 units perform their operation in parallel manner. Thus, the resultant carries(bc16,bc12,bc8 and bc4) and sums(bS[3:0],bS[7:4], bS[11:8] and bS[15:12]) applied to the correction logic at a time. Now, CL using CLA performs the correction operation as shown in Figure 6 and generates the correction values and corresponding carries (dC1, dC2 and dC3), the dC4 is the final carry out of BCD operation.All the adder2 units perform their operation as mentioned in Fig 4in parallel manner. Thus, the resultant sums((dS[3:0], dS[7:4], dS[11:8] and dS[15:12]) are the final BCD sum values.

4. SIMULATION RESULTS

All the proposed designs have been programmed and designed using Xilinx ISE software this software tool provides the two categories of outputs named as simulation and synthesis. The simulation results give the detailed analysis of proposed design with respect to inputs, output byte level combinations. Through simulation analysis of accuracy of the addition, multiplication process estimated easily by applying the different combination inputs and by monitoring various outputs. Through the synthesis results the utilization of area with respect to the programmable logic blocks (PLBs), look up tables (LUT) will be achieved. And also time summary with respect to various path delays will be obtained and power summary generated using the static and dynamic power consumed.

| Name | Value | 999,995 ps | 999,996 ps | 999,997 ps | 999,998 ps | 999,999 ps |
|---------------|-------------|----------------|------------|----------------|------------|------------|
| 🕨 式 sum[15:0] | 00000000000 | | 000 | 0000000010011 | | |
| 🇓 cout | 0 | | | | | |
| 🕨 📷 a[15:0] | 00000000000 | | 000 | 0000000001010 | | |
| 🕨 📷 b[15:0] | 00000000000 | | 000 | 00000000000011 | | |
| 🔚 cin | 0 | | | | | |
| | | | | | | |

Fig 8: simulation output

The above result represents the simulation waveform by using the Xilinx ISE software. Where N is the length of the adder i.e., 16-bit. Finally,corresponding inputs A,B also 16-bit of values 10 and 3,

and cin is 0. Finally, the sum is 32'b 0000 0000 0000 0000 0000 0000 0001 0011, which is in BCD format.

| Device Utilization Summary (estimated values) | | | | | | | | | |
|---|------|-----------|-------------|-----|--|--|--|--|--|
| Logic Utilization | Used | Available | Utilization | - | | | | | |
| Number of Slice LUTs | 42 | 17600 | | 0% | | | | | |
| Number of fully used LUT-FF pairs | 0 | 42 | | 0% | | | | | |
| Number of bonded IOBs | 50 | 100 | | 50% | | | | | |

Fig 9: Design summary

The above result represents the synthesis implementation by using the Xilinx ISE software. From the above table, it is observed that only 42 look up tables are used out of available 17600. It indicates very less area is used for the proposed design.

| Cell:in->out | fanout | Delay | Delay | Logical Name (Net Name) |
|--------------|--------|---------|-----------------|---|
| IBUF:I->0 | 2 | 0.000 | 0.618 | a 2 IBUF (a 2 IBUF) |
| LUT6:I0->0 | 4 | 0.043 | 0.630 | A1/f3/M1/m1 (A1/c2) |
| LUT6:I0->0 | 5 | 0.043 | 0.626 | C1/M6/m (cl<1>) |
| LUT5:10->0 | 2 | 0.043 | 0.410 | A6/f3/m2/out121 (A6/f3/m2/out12) |
| LUT5:13->0 | 8 | 0.043 | 0.642 | C2/M6/m1 (cl<5>) |
| LUT5:10->0 | 2 | 0.043 | 0.410 | A7/f3/m2/out121 (A7/f3/m2/out12) |
| LUT5:I3->0 | 7 | 0.043 | 0.529 | C3/M6/m1 (cl<10>) |
| LUT3:I0->0 | 1 | 0.043 | 0.405 | C4/M6/m3_SW0 (N8) |
| LUT6:14->0 | 4 | 0.043 | 0.620 | C4/M6/m3 (cl<13>) |
| LUT6:I1->0 | 1 | 0.043 | 0.339 | A8/f2/m2/out1 (sum_13_OBUF) |
| OBUF:I->O | | 0.000 | | sum_13_OBUF (sum<13>) |
| Total | | 5.618ns | (0.387 (6.9% | ns logic, 5.231ns route) logic, 93.1% route) |

Fig 10: Time summary

The above result represents the time consumed such as path delays by using the Xilinx ISE software.the consumed path delay is 5.618ns.

| Α | В | С | D | E | F | G | н | I. | J | К | L | М | N |
|-------------------|--------------------|---|---------|------------|---------------|-------------|-----------------|----|----------|-----------|-------------|-------------|-------------|
| Device | | | On-Chip | Power (W) | Used | Available | Utilization (%) | | Supply | Summary | Total | Dynamic | Quiescent |
| Family | Zynq-7000 | | Logic | 0.000 | 34 | 17600 | 0 | | Source | Voltage | Current (A) | Current (A) | Current (A) |
| Part | xc7z010 | | Signals | 0.000 | 75 | | | | Vccint | 1.000 | 0.005 | 0.000 | 0.005 |
| Package | clg400 | | IOs | 0.000 | 50 | 230 | 22 | | Vccaux | 1.800 | 0.006 | 0.000 | 0.006 |
| Temp Grade | Commercial 🗸 | | Leakage | 0.065 | | | | | Vcco18 | 1.800 | 0.001 | 0.000 | 0.001 |
| Process | Typical 🗸 | | Total | 0.065 | | | | | Vccbram | 1.000 | 0.000 | 0.000 | 0.000 |
| Speed Grade | -2 | | | | | | | | Vccpint | 1.000 | 0.020 | 0.000 | 0.020 |
| | | | | | Effective TJA | Max Ambient | Junction Temp | | Vccpaux | 1.800 | 0.013 | 0.000 | 0.013 |
| Environment | | | Thermal | Properties | (C/W) | (C) | (C) | | Vcco_ddr | 1.500 | 0.002 | 0.000 | 0.002 |
| Ambient Temp (C) | 25.0 | | | | 4.0 | 84.7 | 25.3 | | | | | | |
| Use custom TJA? | No 🗸 | | | | | | | | | | Total | Dynamic | Quiescent |
| Custom TJA (C/W) | NA | | | | | | | | Supply | Power (W) | 0.065 | 0.000 | 0.065 |
| Airflow (LFM) | 250 🗸 | | | | | | | | | | | | |
| Heat Sink | Medium Profile 🗸 🗸 | | | | | | | | | | | | |
| Custom TSA (C/W) | NA | | | | | | | | | | | | |
| Board Selection | Medium (10"x10") 🗸 | | | | | | | | | | | | |
| # of Board Layers | 8 to 11 🗸 🗸 | | | | | | | | | | | | |
| Custom TJB (C/W) | NA | | | | | | | | | | | | |

Fig 11: Power summary

The above result represents the power consumed by using the Xilinx ISE software.the consumed power is 0.065w.

| parameter | Conv. BCD | CMOS BCD | BCD | PROPOSEDBCD | | | |
|----------------|-----------|-----------|-----------|-------------|--|--|--|
| | ADDER[4] | ADDER[11] | ADDER[13] | | | | |
| Time delay(ns) | 20.118 | 13.35 | 10.34 | 5.618 | | | |
| Power | 1.293 | 2.356 | 1.46 | 0.065 | | | |
| utilized(uw) | | | | | | | |
| Look up tables | 271 | 562 | 72 | 42 | | | |
| Flip Flops | 237 | 395 | 103 | 59 | | | |
| | | | | | | | |

 Table 1: comparison of 4-bit BCD adder methods



|] | Fig | 12: | comparison | of | various | methods | |
|---|-----|-----|------------|----|---------|---------|--|

508

| Table 2: Performance measureme | nt of BCR ad | der for various | lengths |
|--------------------------------|--------------|-----------------|---------|
|--------------------------------|--------------|-----------------|---------|

| Adder length | bit | Parameter/ method | Slice registers | LUTs | LUT-FF | Delay (nS) | Power (uW) |
|-----------------|-----|--------------------|--------------------|------|--------|------------------------|---------------|
| | | Conv. BCD ADDER[8] | 151 | 282 | 22 | 22 | 65 |
| 0 1.:4 | | CMOS BCD ADDER[11] | 196 | 365 | 18 | 20 | 104 |
| 8-011 | | BCD ADDER[13] | 254 | 471 | 28 | 38 | 90 |
| | | PROPOSED BCD | 55 | 99 | 15 | 11 | 43 |
| | | Conv. BCD ADDER[8] | 243 | 632 | 38 | 35 | 78 |
| | | CMOS BCD ADDER[11] | 342 | 733 | 43 | 30 | 178 |
| 16-bit | | BCD ADDER[13] | 262 | 1011 | 47 | 53 | 115 |
| | | PROPOSED BCD | 142 | 252 | 18 | 17 | 54 |
| | | Conv. BCD ADDER[8] | 853 | 1264 | 69 | 92 | 113 |
| | | CMOS BCD ADDER[11] | 1075 | 1891 | 63 | 133 | 205 |
| 32-bit | | BCD ADDER[13] | 1017 | 2592 | 63 | 180 | 221 |
| | | PROPOSED BCD | 443 | 790 | 36 | 36 | 71 |

From table 1, table 2 and figure 12, it is observed that the proposed BCD adder is area, power and delay efficient compared to the literatures existing method [4],BCD adder [11] and BCD adder [13].

CONCLUSION

We have used QCA technology and the Majority gate technique to construct and describe the operation of a new 32-bit Binary coded Decimal (BCD) adder. The technique is simulated and synthesized using Xilinx ISE software; the results of the simulation and synthesis (implementation) phases have been verified. The results show that the proposed methodology outperforms the existing methods with respect to delay (speed), power usage, area, ADP, and PDP.

REFERENCES

[1] K. Fardad , M. Askari, and M. Taghizadeh, "BCD computing structures in quantum-dot cellular automata," in Proc. International Conference on Computer and Communication Engineering, 2008, pp. 1042-1045.

[2] F. Kharbash and G. M. Chaudhry, "The design of quantum-dot cellular automata decimal adder," in Proc. IEEE International Conference on Multitopic, 2008, pp. 71-75.

[4] G. Cocorullo, P. Corsonello, F. Frustaci, and S. Perri, "Design of efficient BCD adders in quantum-dot cellular automata," IEEE Transactions on Circuits & Systems II Express Briefs, vol. 64, no. 5, pp. 575-579, 2017.

[5] D. Abedi and G. Jaberipur, "Decimal full adders specially designed for quantum-dot cellular automata," IEEE Transactions on Circuits & Systems II Express Briefs, vol. 65, no. 1, pp. 106-110, 2017.

[6] D. Ajitha, K. Ramanaiah, and V. Sumalatha, "An enhanced high-speed multi-digit BCD adder using quantum-dot cellular automata," Journal of Semiconductors, vol. 38, no. 2, pp. 38-46, 2017.

[7] Hafiz Md. Hasan Babu and Ahsan Raja Chowdhury, "Design of a Reversible Binary Coded Decimal Adder by using Reversible 4-bit Parallel Adder ", Proceedings of the 181h International Conference on VLSI Design and 4th International Conference on Embedded Systems Design, 1063-9667/05, IEEE 2005.

[8] M. Mohammadi, M. Eshghi, M. Haghparast, and A Bahrololoom."Design and optimization of reversible bcd adderlsubtractor circuit for quantum and nanotechnology based systems".World Applied Sciences Journal, 4(6):787-792, 2008.

[9] Rahman, Saiflil Islam, Zerina Begum, Hafiz, Mahmud, "Synthesis of Fault Tolerant Reversible Logic Circuits", IEEE, 978-1-4244-2587-7/09,2009.

[10] M. Mohammadi, M. Haghparast, M. Eshghi, and K. Navi. "Minimization optimization of reversible bcd-full adderlsubtractor using genetic algorithm and don't care concept". InternationalJ. Quantum Information, 7(5):969-989, 2009.

[11] Pijush kanti Bhattacharjee, "Digital combinational circuits design by QCA gates", International Journal of Computer and Electrical Engineering, Vol.2 No.1, Feb, 2010, 1793-8163