JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DESIGNING BCD ADDER USING QUANTUM DOT CELLULAR AUTOMATA

[1] K.Srinivasulu, [2] SK.AsmaSulthana, ,M.Tech
[1]M.Tech PG Scholar, [2]Asst.Professor
[1,2]Deptartment Of ECE,
[1,2] SKR College of Engineering and Technology,Konduru satram (v),Manubolu(m),SPSR Nellore(dt),Andhra Pradesh.

Abstract: Due to technology advancement an alternative way is proposed to the classic CMOS technology, that is quantum-dot cellular automata(QCA), which is one of the most prominent solution for designing ultra-low-power and very high speed digital circuits. QCA based implementations are mainly in binary and decimal arithmetic circuits with significant improvements. In this process a new design approach has been presented and demonstrated to design a efficient 4-digit (16 bits) QCA based BCD adders. These advantages become even more evident when two n-digit decimal numbers must be summed. It helps in exploiting innovative logic formulations and purpose designed QCA modules, computational speed significantly higher than existing count parts is achieved without sacrificing either the occupied area or the cell count.

I.INTRODUCTION

The use of decimal arithmetic has been increasing over binary due to increase in the applications of internet banking and there are many others places where precision is very important. Binary digits have a disadvantage of not being able to represent digits like 0.1 or 0.7, requires an infinitely recurring binary number. The availability of multi-operand decimal adders can facilitate financial and commercial applications based on existing huge databases. The simultaneous addition of several decimal numbers is the common operation in multiplication and division algorithms. Multi-operand addition is a vital operation as it is a core component of arithmetic operations, such as division and multiplication. In case of decimal multiplication Multioperand decimal addition comes in handy for swiftly summing large amounts of decimal data. This paper introduces a multi-operand decimal addition algorithm by employing high speed binary to BCD converter circuit, which speeds up the process of decimal addition when multiple BCD operands are added together. A Novel design for 7-bit binary to BCD converter circuit is proposed. Further, analysis is done with respect to the existing binary to BCD converter architectures. The proposed algorithm is fundamentally different from multi-operand BCD addition algorithms since intermediate BCD corrections are not done rather correction is done at the final stage to get proper BCD results. As the decimal corrections are achieved separately from the computation of the binary sum, such that the layout of the binary carry-save adder does not require any further rearrangement.

This project introduces a multi-operand decimal addition algorithm by employing high speed binary to BCD converter circuit, which speeds up the process of decimal addition when multiple BCD operands are added together. A Novel design for 7-bit binary to BCD converter circuit is proposed. The latter allows the carry to be propagated through two subsequent bit-positions with the delay of just one majority gate (MG).

Further, analysis is done with respect to the existing binary to BCD converter architectures. The proposed algorithm is fundamentally different from multi-operand BCD addition algorithms since intermediate BCD corrections are not done rather correction is done at the final stage to get proper BCD results. In the next section, we present preliminary information about the previous work on multi-operand adders and discuss the binary to BCD converter. Hardware implementations normally use BCD instead of binary to manipulate decimal fixed-point operands and integer significand's of DFP numbers for easy conversion.

between machine and user representations. BCD encodes a number X in decimal (non-redundant radix-10) format, with each decimal digit Xi 2 ½0; 9_ represented in a 4-bit binary number system. However, BCD is less efficient for encoding integers than binary, since codes 10 to 15 are unused. Moreover, the implementation of BCD arithmetic has more complications than binary, which lead to area and delay penalties in the resulting arithmetic units.

A variety of redundant decimal formats and arithmetic's have been proposed to improve the performance of BCD multiplication. The BCD carry-save format represents a radix-10 operand using a BCD digit and a carry bit at each decimal position. It is intended for carry-free accumulation of BCD partial products using rows of BCD digit adders arranged in linear or tree-like configurations. Decimal signed-digit (SD) representations.

II. Literature Survey

Decimal arithmetic is the norm in human calculations, and human-centric applications must use a decimal floating-point arithmetic to achieve the same results. Initial benchmarks indicate that some applications spend 50% to 90% of their time in decimal processing, because software decimal arithmetic suffers a 100× to 1000× performance penalty over hardware. The need for decimal floating-point in hardware is urgent. Existing designs, however, either fail to conform to modern standards or are incompatible with the established rules of decimal arithmetic. This paper introduces a new approach to decimal floating-point which not only provides the strict results which are necessary for commercial applications but also meets the constraints and requirements of the IEEE 854 standard. A hardware implementation of this arithmetic is in development, and it is expected that this will significantly accelerate a wide variety of applications

This paper presents a new architecture for a Binary to BCD (BD) converter which forms the core of our Proposed high speed decimal Multi-Operand Adder. Our proposed design contains various improvements over existing architectures. These include an improved BD Converter that helps in reducing the delay of the Multi-operand decimal Adder. Simulation results indicate that with a marginal increase in area, the proposed BD converter exhibits an improvement of 82% in delay over earlier designs. Further the decimal Multi-Operand Adder achieves faster design when compared to previously published results.

The decimal adders and the techniques described herein may be especially suited for numerically intensive commercial applications, such as spreadsheet or financial applications, Industrial Computing where large amounts of decimal data typically need to be processed quickly. When a computer only supports binary arithmetic in hardware, several software packages and programming language extensions for decimal arithmetic have been developed. Multi-operand decimal adders can facilitate financial and commercial applications based on existing huge databases. For Example, Today, BCD data is still heavily used in IBM Processors and databases, such as IBM DB2, mainframes, and power6. all of these are used with in hardware registers and processing units and in software packages.

III.EXISTING SYSTEM

The multi-operand decimal addition is dealt exhaustively by kenney which is carried in serial fashion. The addition is realized by employing CSA's and depending on the carries at the intermediate stages from the CSA's, the design finds decimal correction logic in two ways that are categorized as speculative and non-speculative multi-operand addition. The speculative algorithm adds the correction i.e. six in the intermediate stages while non- speculative does this correction at the later stages of the design based the intermediate CSA carries. In case of Non-speculative, carries are generated as a result of addition of BCD input operands which are passed to the higher significant digit. The final decimal sum and carry is obtained by feeding the sums and carry-outs from the carry-save adder tree into combinational logic network. The architecture utilizes tree of binary CSA`s to compress operands and makes use of a CPA to obtain non-redundant binary value followed by binary to decimal converter cell. This architecture can perform both BCD and binary multi-operand addition

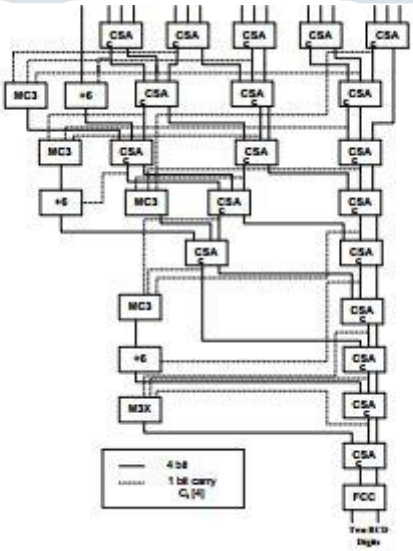


Fig. 3.1.1 Decimal multi-Operand Adder

The paper discusses a design specifically for a 16 Multioperand which does exclusively 16 operand BCD addition, whereas designs can be configured to variable number of BCD operands. The architecture proposes an algorithm on similar lines to that of speculative algorithm proposed by, however performs the computations in parallel. The design introduces new blocks called as Merging three circuit MC3, MC4 and M3X, these
blocks are used for the correction process in the intermediate stages of the architecture followed by Final Correction Circuit (FCC), which produces the BCD.

Binary to BCD conversion The algorithms proposed in converts a 7-bit binary number to 2-digit BCD number to support high performance decimal multiplication. The proposed algorithm in calculates the contributions for lower digit and the higher digit of the BCD number from each of the input binary bits which are later summed in a binary fashion. The contributions are generated as shown in figure 3.1.2 below. The drawback of this algorithm the computation is performed in a binary manner rather than a BCD approach, this results in erroneous BCD conversion. The architecture in is corrected in, by adding the contributions in a BCD fashion. The algorithm in partitions or splits the binary input into two sub-parts, three MSB's and four LSB's. The design then calculates the contributions for the two BCD digits. These contributions are added in BCD fashion to get the final result.

80	40	20	10	8	4	2	1
0	P_6	P_5	P_4	P_3	P_2	P_1	P_0
0	0	P_6	P_5	0	P_4	P_4	0
				0	P_6	P_5	0
b_3	b_2	b_1	b_0	c_3	c_2	c_1	c_0

Fig. 3.1.2 Binary to BCD conversion

Work in proposes two architectures, the Three-Four Split and Four-Three Split binary to BCD converters. The ThreeFour Split algorithms is similar to the one proposed in however designs a clever optimized DL and DH generator blocks, resulting in better performance in terms of area, delay and power

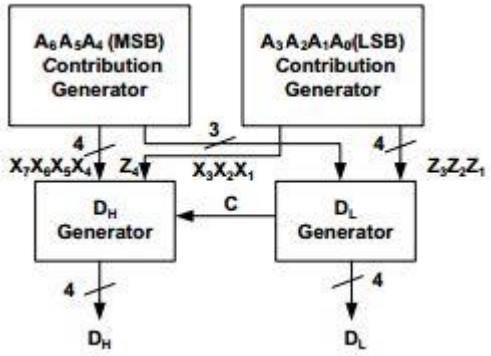


Fig 3.1.3 Three to four split 7-bit BCD convertor

The diagram of the three-four split is depicted in figure 3.1.3. The Four-Three Split design, partitions the 7-bit binary input into four MSB and three LSB bits. Since the LSB bits don't contribute to the higher BCD digit so the LSB contribution generator is removed resulting in area savings at the cost of increase in the complexity of the MSB contribution generator as shown in figure 3.1.4. The Three-Four split is faster than the four- three split whereas the Four-Three Split algorithm results in a more area efficient architecture. In the case of multi-operand addition, the overall delay of adder plays a pivotal role as adders will be in the critical path.

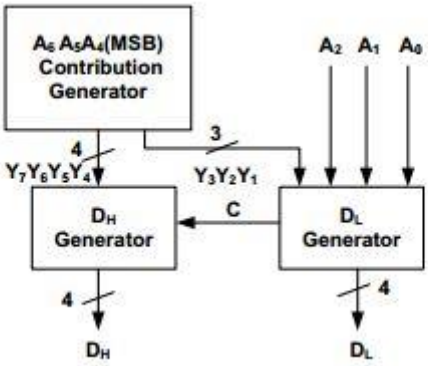


Fig 3.1.4 Four to Three split 7-bit BCD converter

Multi-operand Decimal Adder: The binary parallel multi-operand addition is realized using a CSA tree for compressing the input operands. Efficient multi-operand binary adder circuits can be realized using carry-save adders. The absence of carry propagation until the last stage makes the CSA adders very fast. An added advantage is their simple structure. The proposed algorithm is as depicted in the figure 3.1.4. It comprises of a binary tree structure formed by 3:2 CSA followed by a high speed Binary to Decimal convertor as depicted. The proposed BD converter will be discussed in detail section III B. The proposed multi-operand adder architecture can perform both decimal and binary multi-operand operation. The multi-operand addition of 8 input operands and a carryin (Cin) shown in bold (9+9+9+9+9+9+9+9+7) using the proposed algorithm. We have considered the extreme case of each input operand being 9. Further the carry in from the previous multi-operand column can at most be 7.The algorithm computes the binary sum Sbinary by summing up the input operands in a parallel fashion. Unlike the existing BD converters, we have removed the contribution blocks resulting in a very fast design at the cost of increase in the complexity of the DH and DL generators as shown in figure. The binary output Sbinary is fed to the BD converter which produces 2 digit BCD number, Sdecimal. The design proposed is different from in the sense that the design doesn't do the correction at the intermediate stages, rather it does the decimal conversion after the binary sum has been produced, this improves the performance in terms of speed.

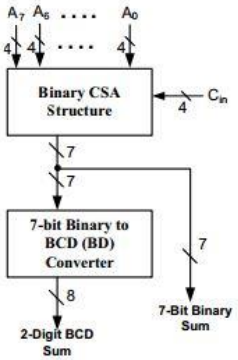


Fig 3.1.5 Multi Operand Decimal/ Binary Adder

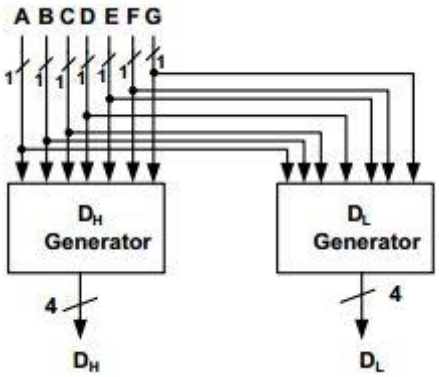


Fig 3.1.6 n-bit BCD converter

Instead of splitting and finding the contributions for DH and DL a method is devised to directly find the contribution to each bit of BCD digits by using equations respectively. The maximum BCD number is nine (910 =10012) so eight operands decimal addition will result in a maximum of seventy-two (7210 = 10010002), hence a carry in to higher position can be maximum of seven. So, the proposed BD convertor block has been optimized to convert a maximum of seventy-nine (7910 =10011112), implying in DH the MSB DH3 can never be `1`. Further, it can be observed that the first two MSB bits of Sbinary cannot be of value `1` simultaneously. Based on this observation the equations (1-2) were devised for the convertor block.

Table 3.1.1 Dh generator of 7-bit BCD converter

j / i	0 D_{H0}	1 D_{H1}	2 D_{H2}	3 D_{H3}
0	A'B'[DEF+C(D+F)'+C'D(E+F)]	B'C[D+E]	0	0
1	B[C'D'+C[DE'+D'(E+F)]]	B[C'D'+CDE]	B[C+D]	0
2	A[D+EF]	AC'	AC'	0

where Pij corresponds to (i , j) element in the Table II The four MSB bits DH0 , DH1 , DH2 and DH3 are formed by replacing j =0 ,1,2,3 in (1). From the Table 3.1.1 and (1) we can obtain the value of DH as follows.

Table 3.1.2 Dl generator of 7-bit BCD converter

i / j	0 D_{L0}	1 D_{L1}	2 D_{L2}	3 D_{L3}
0	G	A'B'[C'[D'F+DEF']+ C[DE'F+D'[E⊕ F]']]	A'B'[CE'[D+F'] + C'E[F+D']]	A'B'[C'DE'F' + C[DEF'+D'E'F]]
1	G	B[D'F'[E+C'] + D[CE'F'+F[E+C']]]	B[C[E'F'D+EFD']+ C'[E⊕ [DF']]]	B[CE'[D⊕ F]'+ C'D'EF]
2	G	A[DF'+C'E'F]	A[E'F+EF'D]	AE[F⊕D]'

3.2 DISADVANTAGES

- The problem associated with complex QCA designs is it uses large amount of white space left between cells is wasted.
- The power consumption is more.
- The delay is more.
- The speed is slow compared to the proposed system.
- Finally, to overcome these problems, the total area covered by QCA cells was minimized by keeping the distance between binary wires as close as possible according to QCA design rules proposed by Kim *et al.*

3.3 PROPOSED SYSTEM

Decimal arithmetic has recently received a great deal of attention since several financial, commercial, and Internetbased applications increasingly require higher precision. In these contexts, the errors coming from the conversion between decimal and binary number representations could not be tolerated, and several recent microprocessors include hardware decimal arithmetic units in their core based on the IEEE 754–2008 standard. The design of such digital circuits requires proper strategies at both logic and layout levels to improve performance and area behaviors. Among the logic circuits recently proposed, arithmetic submodules represent examples of the most investigated structures. Independently of the performed logic function, non-elementary digital modules are designed smartly combining inverters and majority gates (MGs), which are the only fundamental logic gates inherently available within the QCA technology. This brief proposes a novel approach to design QCA-based n-digit (with $n \geq 1$) BCD adders able to achieve computational speed higher than existing counterparts without sacrificing either the occupied area or the cell count. Such advantages are obtained by exploiting an innovative logic strategy together with purpose-designed QCA modules. The latter have been optimized by taking into account that the most time critical decimal addition between two n-digit numbers occurs when a carry is generated by adding the least significant digits of the operands, it is then propagated through the subsequent $n - 2$ digit positions, and finally, it is absorbed at the last digit position where the most significant sum digit is computed. The novel 1-digit decimal adder generates, propagates, and absorbs a carry signal with delay times up to 27%, 45%, and 44% lower than the faster existing counterparts that are those described in. Differently from all previous works, we extended our work to the design of a 2-digit QCA-based BCD adder. Its characterization demonstrates that the 2-digit sum computation is performed within only 18 clock phases, and the circuit spans over an area of only 2.74 μm2. obtained by exploiting an innovative logic strategy together with purpose-designed QCA modules. The latter have been optimized by taking into account that the most time critical decimal addition between two n-digit numbers occurs when a carry is generated by adding the least significant digits of the operands, it is then propagated through the subsequent $n - 2$ digit positions, and finally, it is absorbed at the last digit position where the most significant sum digit is computed. The novel 1-digit decimal adder generates, propagates, and absorbs a carry signal with delay times up to

27%, 45%, and 44% lower than the faster existing counterparts that are those described. Differently from all previous works, we extended our work to the design of a 2-digit QCA-based BCD adder. Its characterization demonstrates that the 2-digit sum computation is performed within only 18 clock phases, and the circuit spans over an area of only 2.74 μm2.

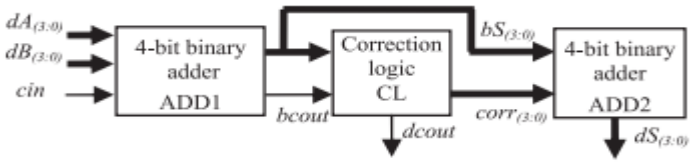


Fig 3.3.1 Structure of the BCD adder

QCA cells are used for both logic structures and interconnections that can exploit either the coplanar cross or the bridge technique. The fundamental logic gates inherently available within the QCA technology are the inverter and the MG. Given three inputs a, b, and c, the MG performs the logic function reported provides that all input cells are associated to the same clock signal clk_x (with x ranging from 0 to 3), whereas the remaining cells of the MG are associated to the clock signal clk_x+1

$M(abc) = a \cdot b + a \cdot c + b \cdot c.$

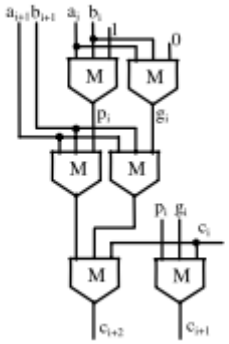
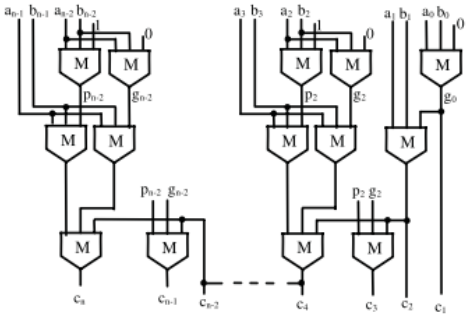


Fig 3.3.2 Novel 2-bit basic module

Several designs of adders in QCA exist in literature. The RCA and the CFA process n-bit operands by cascading n full-adders (FAs). Even though these addition circuits use different topologies of the generic FA, they have a carry-in to carry-out path consisting of one MG, and a carry-in to sum bit path containing two MGs plus one inverter. As a consequence, the worst case computational paths of the n-bit RCA and the n-bit CFA consist of (n+2) MGs and one inverter.

A CLA architecture formed by 4-bit slices was also presented in. In particular, the auxiliary propagate and generate signals, namely pi = ai + bi and gi = ai · bi, are computed for each bit of the operands and then they are grouped four by four. Such a designed n-bit CLA has a computational path composed of 7 + 4 × (log4 n) cascaded MGs and one inverter. This can be easily verified by observing that, given the propagate and generate signals (for which only one MG is required), to compute grouped propagate and grouped generate signals; four cascaded MGs are introduced in the computational path. In addition, to compute the carry signals, one level of the CLA logic is required for each factor of four in the operands word-length. This means that, to process n-bit addends, log4 n levels of CLA logic are required, each contributing to the computational path with four cascaded MGs. Finally, the computation of sum bits introduces two further cascaded MGs and one inverter.

The parallel-prefix BKA demonstrated in exploits more efficient basic CLA logic structures. As its main advantage over the previously described adders, the BKA can achieve lower computational delay. When n-bit operands are processed, its worst case computational path consists of 4 × log2 n − 3 cascaded MGs and one inverter. Apart from the level required to compute propagate and generate signals, the prefix tree consists of 2 × log2n −2 stages. From the logic equations provided in, it can be easily verified that the first stage of the tree introduces in the computational path just one MG; the last stage of the tree contributes with only one MG; whereas, the intermediate stages introduce in the critical path two cascaded MGs each. Finally, for the computation of the sum bits, further two cascaded MGs and one inverter are added.



(a)

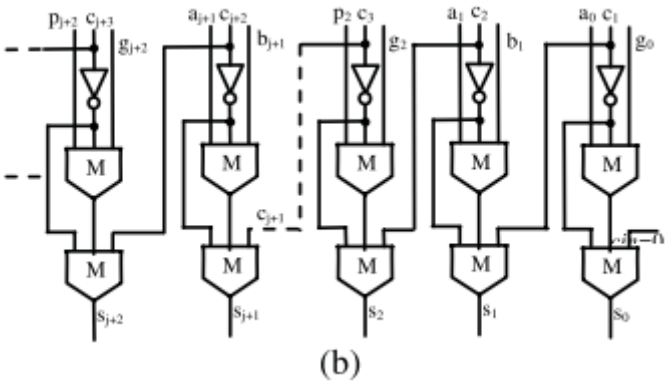
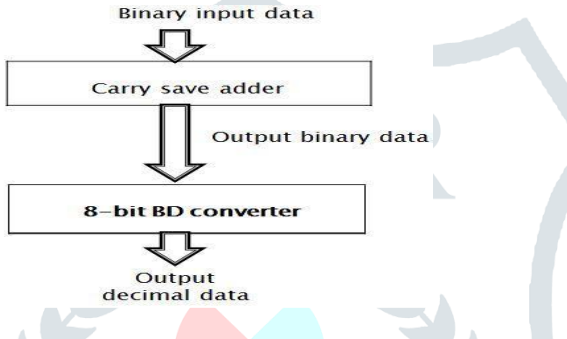


Fig 3.3.3 Novel n-bit adder (a) carry chain and (b) sum block.

IMPLEMENATION

Binary n-digit input data is given to the carry save adder. He carry save adder is capable of taking three inputs and results in n-bit sum and n+1 bit carry. The output of the carry save adder is a binary data. The binary data can be converted into decimal data by using a 8-bit bcd converter.

Fig 3.3.2.1: General Block diagram of BCD Adder



$A_0 = 9 = 1001$, $A_1 = 9 = 1001$, $A_2 = 9 = 1001$ → $S_1 = 1001$, $C_1 = 10010$

$A_3 = 9 = 1001$, $A_4 = 9 = 1001$, $A_5 = 9 = 1001$ → $S_2 = 1001$, $C_2 = 10010$

$A_6 = 9 = 1001$, $A_7 = 9 = 1001$, $A_8 = 9 = 1001$ → $S_3 = 1001$, $C_3 = 10010$

$A_9 = 9 = 1001$, $A_{10} = 9 = 1001$, $A_{11} = 9 = 1001$ → $S_4 = 1001$, $C_4 = 10010$

$A_{12} = 9 = 1001$, $A_{13} = 9 = 1001$, $A_{14} = 9 = 1001$ → $S_5 = 1001$, $C_5 = 10010$

$A_{15} = 9 = 1001$, $S_1 = 1001$, $S_2 = 1001$ → $S_6 = 1001$, $C_6 = 10010$

$S_3 = 1001$, $S_4 = 1001$, $S_5 = 1001$ → $S_7 = 1001$, $C_7 = 10010$

$C_1 = 10010$, $C_2 = 10010$, $C_3 = 10010$ → $S_8 = 10010$, $C_8 = 100100$

$C_4 = 10010$, $C_5 = 10010$, $C_6 = 10010$ → $S_9 = 10010$, $C_9 = 100100$

$S_6 = 01001$, $S_7 = 01001$, $S_8 = 10010$ → $S_{10} = 10010$, $C_{10} = 010010$

$C_7 = 10010$, $C_8 = 10010$, $C_9 = 10010$ → $S_{11} = 010010$, $C_{11} = 1001000$

$S_9 = 010010$, $S_{10} = 010010$, $S_{11} = 010010$ → $S_{12} = 010010$, $C_{12} = 0100100$

$C_{10} = 0010010$, $C_{11} = 1001000$, $C_{12} = 0100100$ → $S_{13} = 01111110$, $C_{13} = 00000000$

$S_{12} = 00010010$, $S_{13} = 01111110$, $C_{13} = 00000000$ → $S = 01101100$, $C = 000100100$

$S = 01101100$, $C = 000100100$ → $S_{binary} = 010010000$ → 9-bit BD Converter → $S_{decimal} = 0001\ 0100\ 0100$ ($D_3\ D_2\ D_1$)

The 16- bit Binary to BCD Converter: The proposed BD converter was designed with the intention of speeding up of the multi-operand decimal adder. The n-bit binary number is converted to the two BCD digits i.e. four LSB bits and four MSB bits of Sdecimal form DL and DH respectively. In the contribution block adds to the critical path delay of the converter, the proposed design aims at removing these contribution blocks and thereby drastically improving the overall delay of the converter.

Fig 3.3.2.2 Example of proposed sixteen operands decimal addition

3.4 ADVANTAGES

- High speed
- Low power consumption
- Lesser area

3.5 Feasibility Study

Feasibility study is accompanied once the difficult is obviously understood. The feasibility study which is a great level lozenge version of the whole system analysis and design procedure. The independent is to define whether the planned system is possible or not and it benefits us to the least expense of how to resolve the problem and to govern, if the Problem is wealth solving.
The following are the three important tests that have been conceded out for feasibility Study.

- Technical Feasibility
- Economic feasibility
- Operational feasibility

3.5.1 Technical Feasibility

In the technical feasibility study, one has to assess whether the implemented system can be established using existing technology or not. It is intended to implement the implemented system in JSP. The project enabled is theoretically feasible since the following reasons.

- All needed technology exists to improve the system.
- The existing system is so malleable that it can be advanced further.

3.5.2 Economic Feasibility

As a portion of this, the expenses and profits related with the implemented systems are to be associated. The project is carefully feasible only if tangible and intangible assistances balance the cost. We can say the implemented system is feasible founded on the following grounds.

- The charge of developing the filled system is sensible.
- The cost of hardware and software for the application is less.

3.5.3 Operational Feasibility

This project is operationally feasible for there is necessary support from the project organization and the users of the implemented system Implemented system absolutely does not damage and determination not create the corrupt results and no problem will ascend after implementation of the system.

IV.Results

Block Diagram

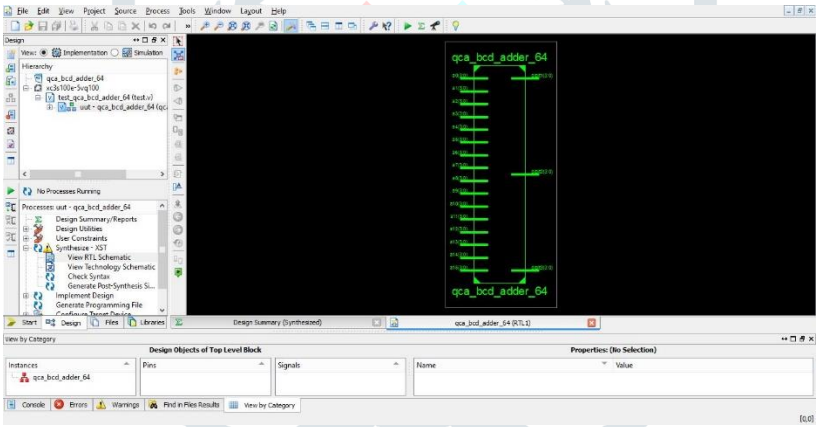


Figure 4.1: Block Diagram of the QCA BCD adder 64bit

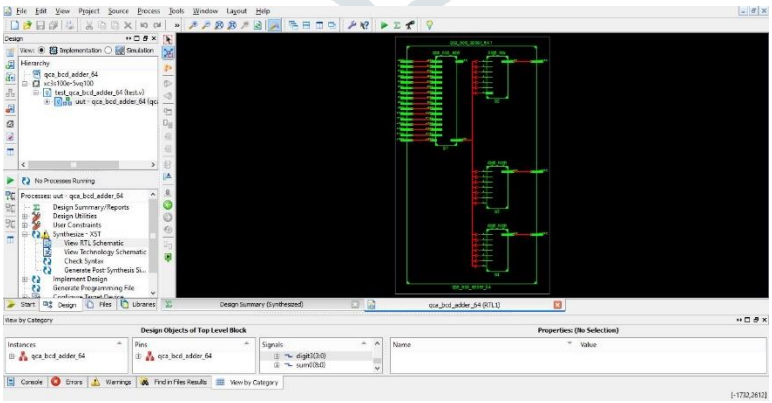


Figure 4.2: RTL Schematic of the QCA BCD adder 64bit

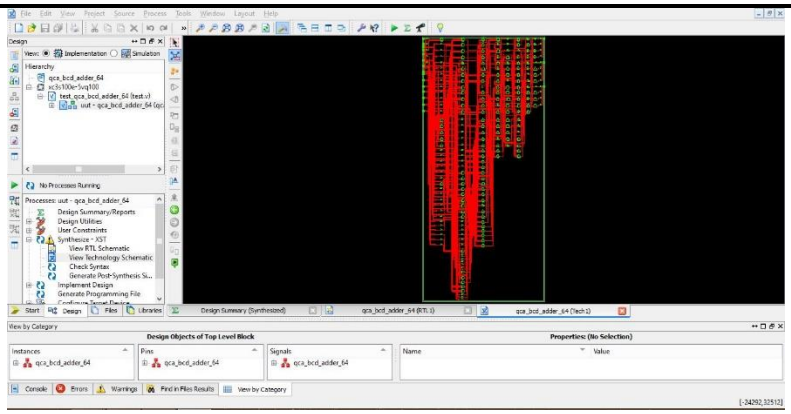


Figure 4.3: Technology Schematic of the QCA BCD adder 64bit

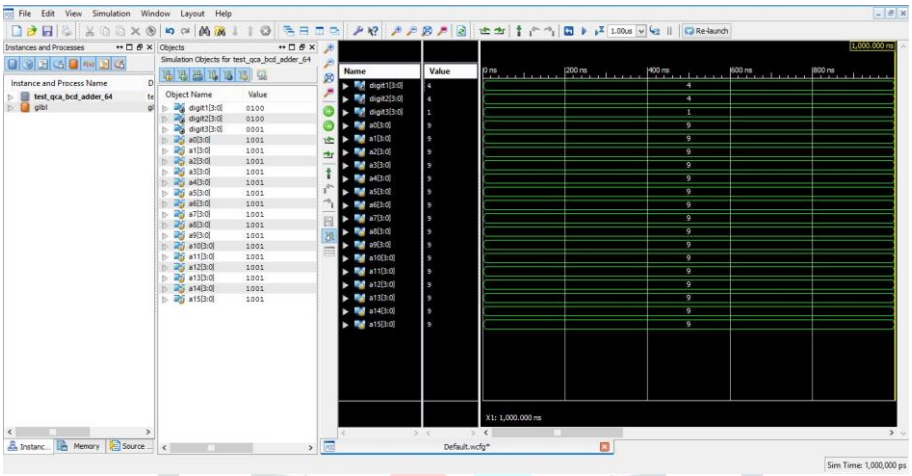


Figure 4.4: Simulation result of the QCA BCD adder 64bit

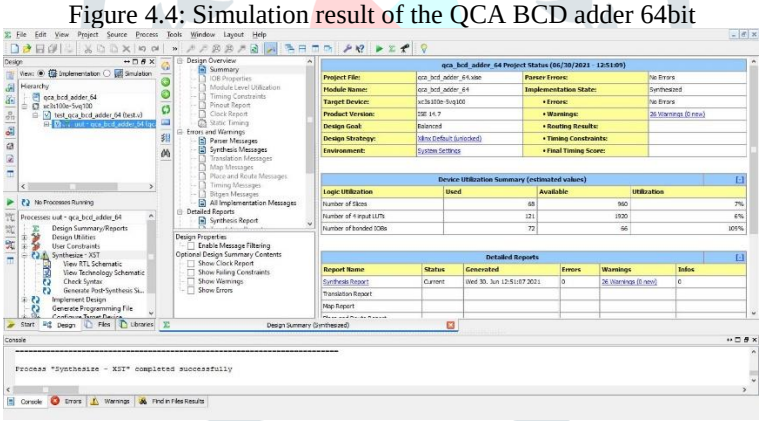


Figure 4.5: Summary report of the QCA BCD adder 64bit

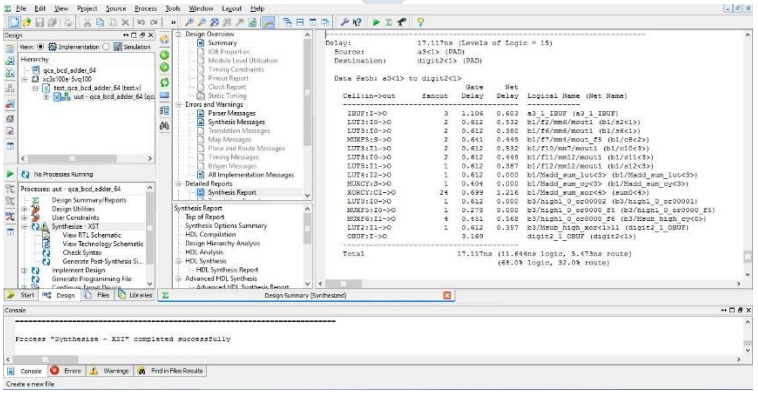


Figure 4.6: Delay report of the QCA BCD adder 64bit

Comparison Table

Table-6.1 Area and delay comparison between BCD adders and QCA adders

	Area in LUT	**Delay in ns**
64-bit BCD Adder	127	19.162 ns
64-bit QCA Adder	121	17.117 ns

V.Conclusion

A new design approach has been presented and demonstrated to achieve efficient QCA-based implementations of decimal adders. Unconventional logic formulations and purposed designed logic modules here proposed allow outperforming decimal adders known in the literature. In fact, the new 1-digit BCD adder exhibits computational delay and area occupancy up to 36% and 52% lower than existing competitors. These advantages become even more evident when two n-digit decimal numbers must be summed. The 2-digit adder designed as proposed here operates within only 18 clock phases and occupies an area of just 2.74 μm2. Finally, by exploiting the 2-D wave clocking scheme, a more feasible implementation of the new adder has been obtained without compromising the advantages achieved over its direct competitors.

VI.REFERENCES

- M. F. Cowlishaw. Decimal floating-point: Algorism for computers. In Proc. IEEE 16th Symposium on Computer Arithmetic, pages 104–111, July 2003.
- M. D. Ercegovac and T. Lang, Digital Computer Arithmetic. Elsevier/Morgan Kaufmann Publishers,2004.
- R. D. Kenney and M. J. Schulte. High-speed multi-operand decimal adders. IEEE Trans. on Computers, 54(8):953–963,Aug.2005.
- Dadda, Luigi. "Multi-operand parallel decimal adder: A mixed binary and bcd approach." Computers, IEEE Transactions on 56.10 (2007): 1320-1328.
- Lin, Kuan Jen, et al. "A parallel decimal adder with carry correction during binary accumulation." New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International. IEEE, 2012.
- Jaberipur, Ghassem, and Amir Kaivani. "Improving the speed of parallel decimal multiplication." Computers, IEEE Transactions on 58.11 (2009): 1539-1552.
- Bhattacharya, Jairaj, Aman Gupta, and Anshul Singh. "A high performance binary to BCD converter for decimal multiplication." VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on. IEEE, 2010.
- Al-Khaleel, Osama, et al. "Fast and compact binary-to-BCD conversion circuits for decimal multiplication." Computer Design (ICCD), 2011 IEEE 29th International Conference on. IEEE, 2011.
- S. Knowles, ‘‘A family of adders,’’ in: Proceedings of the 14th IEEE Symposium on Computer Arithmetic, pp. 30–34, 1999.