DESIGN OF DECIMAL / BINARY MULTI-OPERAND ADDER USING A FAST BINARY TO DECIMAL CONVERTER

SK. Howldar[1], M. Vamsi Krishna Allu [2]

[1] M.Tech VLSI Design student, [2]Assistant Professor

[1,2]E.C.E Department, Sir C. R. Reddy college of Engg, Andhra University

ABSTRACT

The new architecture that is proposed is for a 7-bit 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 7-bit BD Converter that helps in reducing the delay of the Multi-operand decimal Adder. Simulation results indicate shows that we can get out put both in binary & BCD for multi operand addition with a marginal increase in area, the proposed BD converter exhibits an improvement over the traditional architectures. Overall Power Delay Product is improved

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.

Decimal data processing applications have grown exponentially in recent years thereby increasing the need to have hardware support for decimal arithmetic. Binary to BCD conversion forms the basic building block of decimal digit implemented using iterative approaches or lookup table based reduction schemes. adders. Decimal arithmetic operations are generally slow and complex, its hardware occupies more area. They are typically

This has led to the motivation behind improving BCD architectures, to enable faster and compact arithmetic.

II. BACKGROUND

A. Multi-operand Addition

The multi-operand decimal addition is dealt exhaustively by kenney [3] 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 [3]. 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 [4] 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.

B. Binary to BCD conversion

This algorithm is to perform highly efficient fixed bit binary to BCD conversion in terms of delay, power and area.

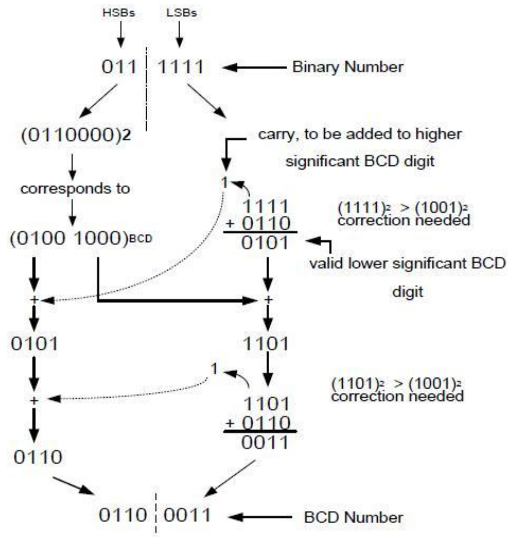

HSBs
LSBs
011
1111
Binary Number
(0110000)2
corresponds to
(0100 1000)BCD
carry, to be added to higher significant BCD digit
1
1111
+ 0110
0101
(1111)2 > (1001)2 correction needed
valid lower significant BCD digit
+
+
0101
1101
1
(1101)2 > (1001)2 correction needed
1101
+ 0110
0011
+
0110
0110
0011
BCD Number

Fig1 algorithm for number (111111)2 or (0110 0011) BCD

Let p6p5p4p3p2p1p0 be the seven binary bits to be converted into two BCD digits. To convert these binary bits into 2-digit BCD we split the binary number into two parts, the first part contains the lower significant bits (LSBs) p3, p2, p1 and p0 while the second part contains the remaining higher significant bits (HSBs) p6, p5 and p4.

The lower significant part (LSBs) has the same weight as that of a BCD digit and can be directly used to represent a BCD digit. The only exception arrives when p3p2p1p0 exceeds (1001)2 or (9)10. To convert the LSBs into a valid BCD number we check whether p3p2p1p0 exceeds (1001)2, and if it does, we add (0110)2 to it. This procedure of adding (0110)2 whenever the number exceeds (1001)2 is called correction in BCD arithmetic. The carry obtained from this procedure is added to the higher significant BCD digit calculated from the HSBs of the original binary number. The HSBs not only contribute to the higher significant BCD digit but also to the lower significant BCD digit. These contributions of HSBs towards the lower significant digit are added after BCD correction. The resulting sum is then checked for the case (1001)2 and correction is done if needed to obtain the final lower significant BCD digit. A possible carry from the above operation is added to the higher significant digit resulting in the final higher significant BCD digit. Figure 1 shows an example of the algorithm for number $(111111)_2$ or $(63)_{10}$ or $(0110\ 0011)_{BCD}$.

III. PROPOSED DESIGN

A. 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 [9]. An added advantage is their simple structure. The proposed algorithm is as depicted in the figure 2. It comprises of a binary tree structure formed by 3:2 CSA followed by a high speed Binary to Decimal convertor as depicted in figure 2. The proposed BD converter will be discussed in detail section III B.

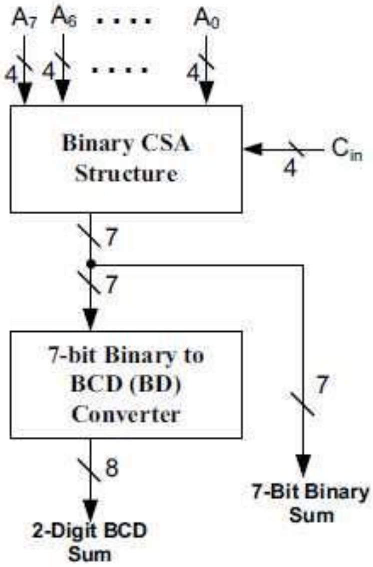


Fig.2. Proposed multi-operand decimal/binary adder

The proposed multi-operand adder architecture can perform both decimal and binary multi-operand operation. Figure 3 illustrates the multi-operand addition of 8 input operands and a carry in (C_{in}) shown in bold (9+9+9+9+9+9+9+9+7) using the proposed algorithm in figure 2. 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 [7-8] 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 [5] 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.

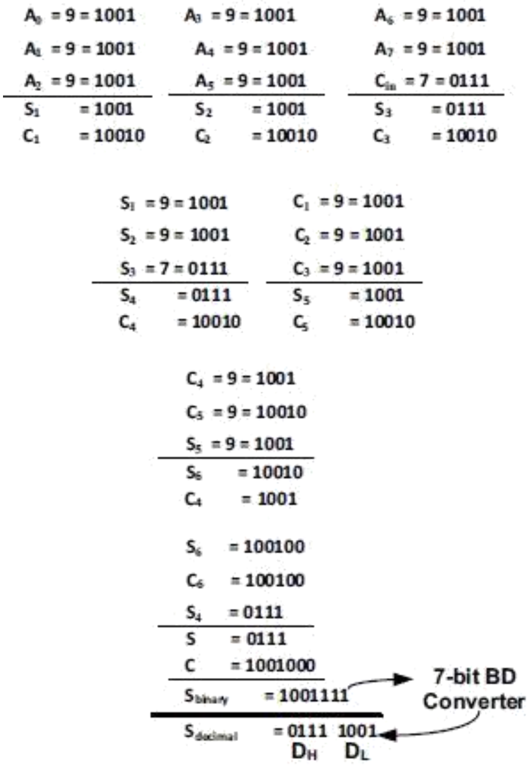


Fig. 3 Example of Proposed 8-operand decimal addition

B. Proposed 7- bit Binary to BCD Converter

The proposed BD converter was designed with the intention of speeding up of the multi-operand decimal adder. The 7-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 [8] 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 as depicted in figure 4.

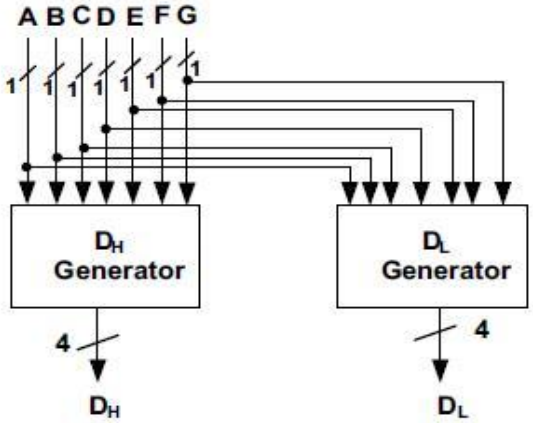


Fig.4 Proposed 7-bit BD converter

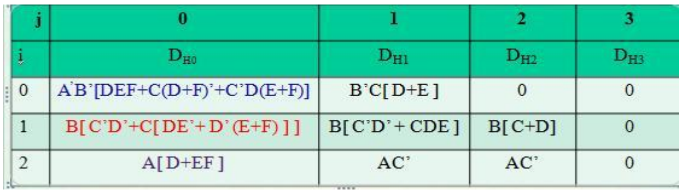
Instead of splitting and finding the contributions for DH and DL a method is devised to directly find the contribution to each bit of BCD digits by using equations (1) and (2) respectively.

The maximum BCD number is nine (910 =10012) so eight operand 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.

$$D_{Hj} \;\square\; \sum_{i \square 0}^{2} P_{ij}$$

TABLE I. PROPOSED DH GENERATOR OF 7-BIT BD CONVERTER



j	0	1	2	3
i	D_{H0}	D_{H1}	D_{H2}	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 I and (1) We can obtain the value of DH as follows.

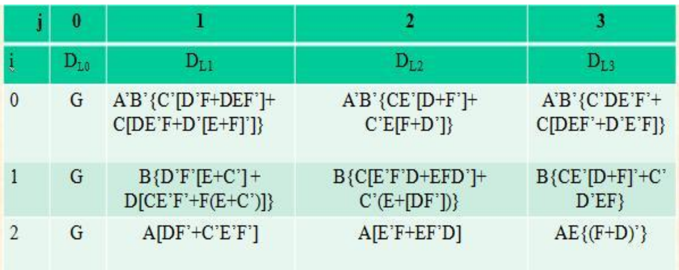
CASE 1: j=0 then

D_{H0} =A' B'[DEF+C(D+F)'+C'D(E+F)] + B[C'D'+C[DE'+ D' (E+F)]] + A[D+EF]

Similarly we can obtain the values of DH1, DH2, DH3 by replacing the value of j = 1, 2, 3 in (1)

$$D_{Lj} \;\square\; \sum_{i \square 0}^{2} Q_{ij}$$

TABLE II. PROPOSED DL GENERATOR OF 7-BIT BD CONVERTER



j	0	1	2	3
i	D_{L0}	D_{L1}	D_{L2}	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)'}

Where Qij corresponds to (i , j) element in the Table -II. The four LSB bits DL0 , DL1 , DL2 , DL3 are produced by substituting j =0,1,2,3 in (3). From the Table II and (4) we can achieve the value of four bits of DL as follows.

Case I: j = 0

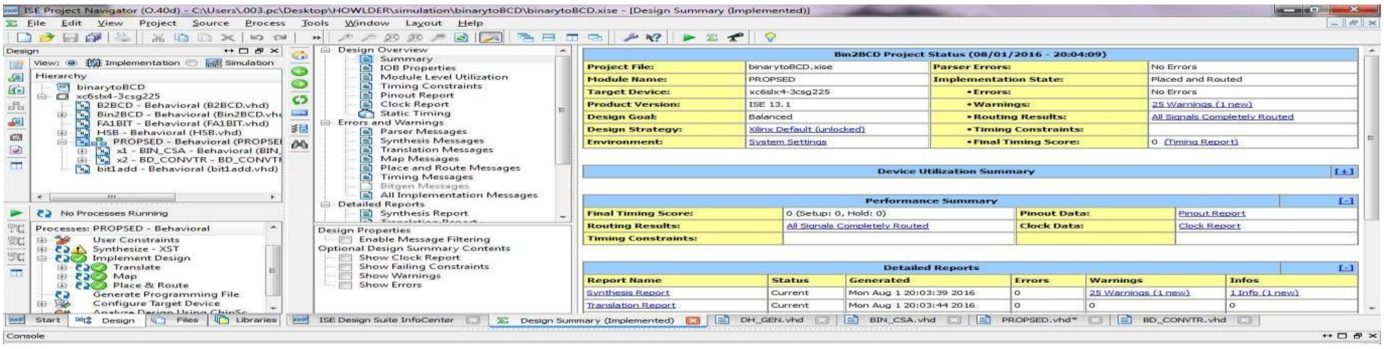
DL0 □G

On similar lines we can obtain the values of DL1, DL2, DL3 By substituting the value of j = 1 ,2, 3 in (3)

IV. SIMULATION RESULTS

The simulation results for 8 operand decimal binary addition a0 to a7 with four bit size, output is in both Binary & BCD form. Fig 5 shows Sythesis report Fig 6 Simulation results Fig 7 shows RTL schematic

PARAMETERS	**PROPSED DESIGN**	**New Design (using 3-4 split)**
Area (µm2)	242.244	150.293
Delay (ns)	12.247	22.36
Power (uW)	1403.1	1106.22
Power Delay Product (fJ)	1.7182541	2.4668706



ISE Project Navigator (O.40d) - C:\Users\.003.pc\Desktop\HOWLDER\simulation\binarytoBCD\binarytoBCD.xise - [Design Summary (Implemented)]

Bin2BCD Project Status (08/01/2016 - 20:04:09)

Project File:	binarytoBCD.xise	**Parser Errors:**	No Errors
Module Name:	PROPSED	**Implementation State:**	Placed and Routed
Target Device:	xc6slx4-3csg225	**• Errors:**	No Errors
Product Version:	ISE 13.1	**• Warnings:**	25 Warnings (1 new)
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary [+]

Performance Summary [-]

Final Timing Score:	0 (Setup: 0, Hold: 0)	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:			

Detailed Reports [-]

Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Mon Aug 1 20:03:39 2016	0	25 Warnings (1 new)	1 Info (1 new)
Translation Report	Current	Mon Aug 1 20:03:44 2016	0	0	0

FIG 5 SHOWS SIMULATION RESULTS

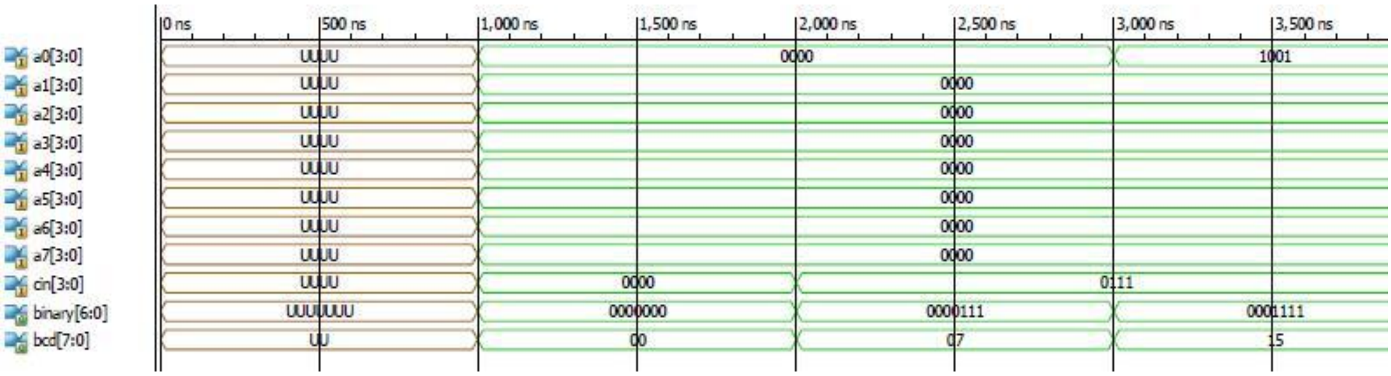


FIG 6 SHOWS SIMULATION RESULTS

V. CONCLUSIONS

A Novel Unified BCD/ Binary multi-operand addition algorithm has been proposed. The binary parallel multioperand addition is realized using a CSA tree for compressing the input operands. The proposed BD converter forms the core of the multi-operand decimal adder. Simulation results demonstrate the efficiency of our proposed BD converter as well as multi-operand decimal adder with respect to exiting designs.

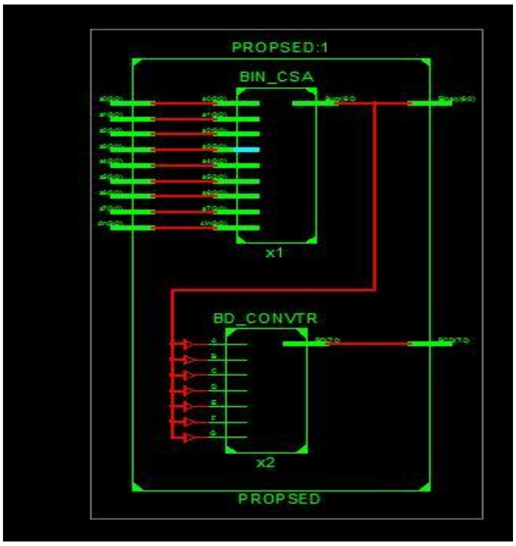


Fig7 RTL schematic

REFERENCES

[1] M. F. Cowlishaw. Decimal floating-point: Algorism forcomputers. In Proc. IEEE 16th Symposium on Computer Arithmetic,pages 104–111, July 2003.

[2] M. D. Ercegovac and T. Lang, Digital Computer Arithmetic.Elsevier/Morgan Kaufmann Publishers,2004.

[3] R. D. Kenney and M. J. Schulte. High-speed multi-operand decimal adders. IEEE Trans. on Computers, 54(8):953–963,Aug.2005.

[4] Dadda, Luigi. "Multi-operand parallel decimal adder: A mixedbinary and bcd approach." Computers, IEEE Transactions on 56.10(2007): 1320-1328.[5] Lin, Kuan Jen, et al. "A parallel decimal adder with carrycorrection during binary accumulation." New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International. IEEE, 2012.

[5] [6]- Jaberipur, Ghassem, and Amir Kaivani. "Improving the speed ofparallel decimal multiplication." Computers, IEEE Transactions on 58.11 (2009): 1539-1552.

[6] Bhattacharya, Jairaj, Aman Gupta, and Anshul Singh. "A highperformance binary to BCD converter for decimal multiplication." VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on. IEEE, 2010. [8]- 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.

[7] S. Knowles, ‘‘A family of adders,’’ in: Proceed ings of the 14th IEEE Symposium on Computer Arithmetic,

pp. 30–34, 1999.

AUTHORS PROFILE

SK.HOWLDAR Received his B.E degree in Electronics and Instrumentation Engineering from Sir.C.R.Reddy College of Engineering, affiliated to ANDHRA UNIVERSITY, Vishakapatnam. Currently he is persuing M.E degree in VLSI Design from Sir. C. R. Reddy College of Engineering, Eluru.

M. VAMSI KRISHNA ALLU Received M. Tech degree in **VLSI Design** from Sir C. R. Reddy college of Engineering, affiliated to ANDHRA UNIVERSITY, Visakhapatnam. Currently he is an assistant professor in the Department of Electronics and Communication Engineering, Sir. C. R. Reddy College of engineering, Eluru. His area of interest in Low Power VLSI Design He is a life member of ISTE and Institute of Engineers (IE).