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

Area Efficient High Speed BCD Adders using Modified Carry Select Adder

[1]Ayushi Agrawal, [2]Dr. Prashant Chaturvedi
[1]Research Scholar, [2]Professor
[1&2] Department of Electronics and Communication Engineering,
[1&2] Lakshmi Narain College of Technology, Bhopal, India

Abstract— The demand for massive computation increases day by day. It leads to the requirement of high- speed adders for arithmetic operations. The high-speed adder consumes more area and power. The aim of this paper is to achieve low area of high speed adder without increasing the delay of the adder. The conventional BCD adder is very slow because of the huge carry chain delay. Therefore, a high-speed BCD adder with a KSA is proposed. In this design, a new detector circuit has been employed to generate the carry-out of each stage in two gate delays. This proposed BCD adder is faster than other existing BCD adders, but the area also rises. The binary to decimal converter was designed with new formulae so that it directly converts binary partial products to equivalent decimal number.

Keywords— ***Kogge Stone Adder, Binary Coded Decimal (BCD), Delay***

I. INTRODUCTION

A binary number is expressed in base-2 number system, which has '0'and '1'only. In the binary system, each digit is called a binary digit (bit). The binary numbers are represented as binary numbers without fractional part (Whole numbers) and with a fractional part [1]. The latter representation can be further classified as fixed-point and floating-point binary numbers. A decimal number system is a base-10 number system. This decimal number system is also classified as binary number system. This number system is used as input data by human beings in most of the places or applications. Binary Coded Decimal (BCD) is one of the ways which is used to represent the decimal numbers in binary form. In BCD, each digit expressed in four bits. The advantage of BCD is an easy way of representing decimal numbers [2].

The paired numbering framework is, by a long shot, the most normal numbering framework being used in PC frameworks today. In days long, notwithstanding, there were PC frameworks that depended on the decimal (base 10) numbering framework as opposed to the parallel numbering framework. Such PC frameworks were extremely well known in frameworks focused on for business/business frameworks. In spite of the fact that frameworks architects have found that twofold number juggling is quite often better than decimal math for general estimations, the fantasy actually perseveres that decimal number juggling is better for cash estimations than paired number juggling. Hence, numerous product frameworks actually indicate the utilization of decimal number juggling in their estimations [3].

BCD portrayal offers one major benefit over paired portrayal: it is genuinely paltry to convert between the string portrayal of a decimal number and its BCD portrayal. This highlight is especially gainful while working with fragmentary qualities since fixed and drifting point paired portrayals can't by and large address many regularly utilized values somewhere in the range of nothing and one (e.g., 1/10). Subsequently, BCD activities can be proficient while perusing from a BCD gadget, doing a basic number-crunching activity (e.g., a solitary expansion) and afterward composing the BCD worth to some other gadget. Numerous designs and calculations have been proposed to date for decimal math [4].

It planned a viper for repetitive BCD expansion. However the plan includes basic change of a BCD number to repetitive BCD and perform expansion in excess structure and again convert the outcome back to BCD structure, it experiences high postponement .what's more the change hardware utilized adds to the plan intricacy. Calculations and adders for BCD expansion are introduced in which the plan utilizing speculative expansion strategy have ordinary design and their remedy unit is autonomous of the quantity of input operands while the other plan utilizing non-speculative expansion have lower delays [5, 6].

II. BCD ADDERS

A binary adder can be used for adding two binary numbers (whole number or fixed point number), and this binary adder is used to sum up, the trailing significand of inputs of a binary floating-point adder. Hence it is a part of a binary floating-point adder also. Let us take an example. There are two decimal numbers A and B. The (decimal) value of A and B is 15.4 and 5.5, respectively. The summation of the numbers has to be determined. Now, this operation can be performed in two ways: (i) using binary adders alone (ii) using BCD adders. First, we consider that the addition operation is performed by binary adders solely. Initially, these numbers are converted to binary numbers. Assume A and B are represented in a 12-bit binary number, including four bits for fractions. The fraction of A 0.4 cannot be represented exactly in binary because 0.4 gives a nonterminating binary. Hence, 15.375 is assigned to A instead of 15.4. Consequently, there is a small difference in the results which is shown in Fig.1.1.a For adding decimal numbers A and B, using BCD adder, first, these numbers are converted to BCD numbers. Here, A and B are represented in a 12-bit BCD number with a 4-bit decimal point. A and B are ternary numbers. They are not like the previous method. As a result, the exact output is obtained, as shown in Fig.1.1.b. The main advantage is that the accurate result is obtained, but the area, delay and power consumption of the BCD adder are more than the binary adder. Though the binary floating point adder is used for the above problem, the obtained result does not exactly match with the expected results. From the above facts, We can say that the BCD adder is better than using binary adder alone for adding decimal numbers with a fraction due to the exact result.

A = 15.375 (Decimal) = 0000 1111.0110 (Binary)

B = 05.500 (Decimal) = 0000 0101.1000 (Binary)

Result = 20.875 (Decimal) = 0001 0100.1110(Binary) (Obtained)

Result = 20.9 (Decimal) (Expected)

(a)

A = 15.4 (Decimal) = 0001 0101.0100 (BCD)

B = 05.5 (Decimal) = 0000 0101.0101 (BCD)

Result = 20.9 (Decimal) = 0010 0000.1001 (BCD) (Obtained)

Result = 20.9 (Decimal) (Expected)

(b)

Fig. 1: Addition of fixed point decimal numbers: (a)using binary adders (b)using BCD adder

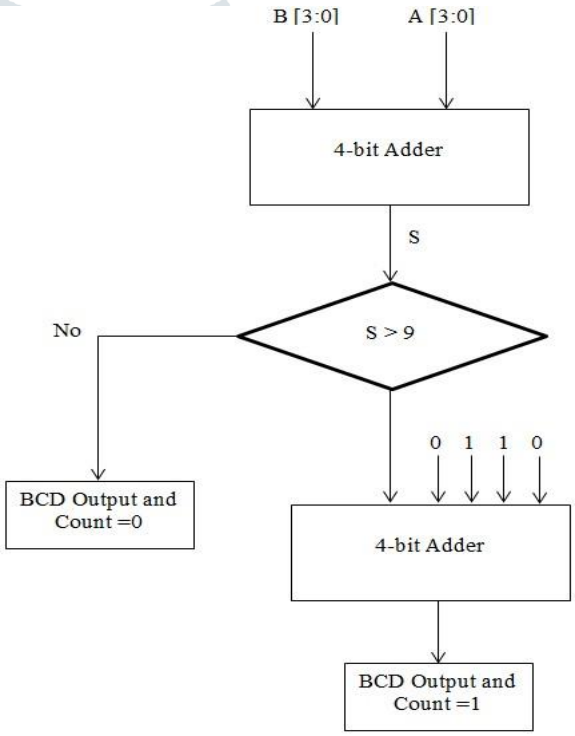
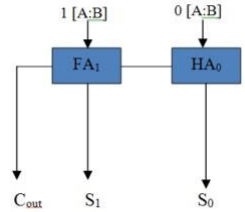
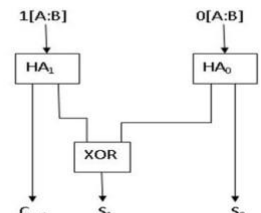


Fig. 2: Flow Chart of Proposed Methodology

Kogge Stone Adder

Eventually, all the designing levels of digital system or IC's Packages depend on number of gates in a single chip that is also called bottom up approach.

Table 1: Comparison Kogge Stone Adder and Modified Kogge Stone Adder

Existing Architecture	Proposed Architecture
 Figure 3 (a): A Modified 2 bit KS Adder	 Figure 3 (b): A Modified 2 bit MKS Adder
2-bit Kogge Stone adder consist of three half adder and one OR Gate.	2-bit modified Kogge Stone adder consist of two half adder and one XOR Gate.
Increase the bit of Kogge Stone adder is more complexity.	Increase the bit of modified Kogge stone adder is less complexity compare to Kogge Stone adder.
Increase the bit of Kogge Stone adder is increase the delay.	Increase the bit of modified Kogge Stone adder is less delay compare to Kogge Stone adder.

III.

six

PROPOSED TECHNOLOGY

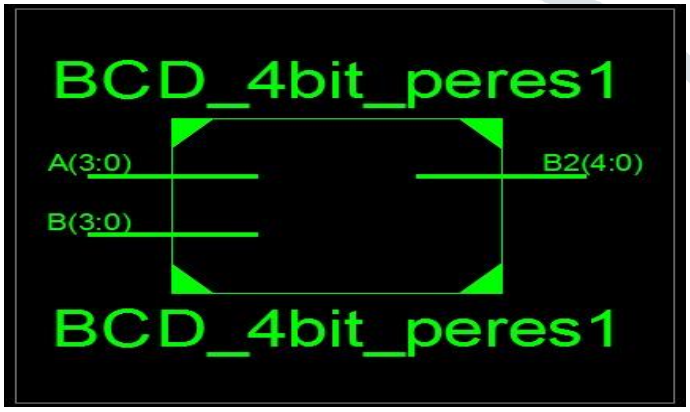
BCD binary numbers represent Decimal digits 0 to 9. A 4bit BCD code is used to represent the ten numbers 0 to 9. Since the 4-bit Code allows 16 possibilities, therefore the
first 10 4-bit combinations are considered to be valid BCD

$$S_i = P_i \quad (1)$$

View of 4-bit Reversible BCD Adder using

combinations. The latter combinations are invalid and do not occur BCD Code has applications in Decimal Number display System such as counters and digital clock. BCD numbers can be added together using BCD addition.

BCD addition is similar to normal binary addition except for the case when sum of two BCD digits exceeds 9 or a carry is generated. When the sum of two BCD numbers exceeds 9 or a carry is generated a 6 is added to convert the invalid number into a valid number. The carry generated by adding a 6 to the invalid BDC digit is passed on to the next BCD digit.



Modified KS adder can be reduced regarding the area or number of gates. If we remove the first XOR gate from modified KS adder nothing will be changed for result but area and propagation delay will be reduced.

Fig. 5: RTL

This equation is applied only for first summation output. For the next sum bit equation will be changed.

$$S_i = P_i \oplus G_{i-1} \quad (2)$$

In this modified Fig. 3 excluded the first XOR gate. The sum bit of the first half adder shows the LSB bit of the output and carry output will connect to the first pin of XOR gate. The sum bit of the second half adder is connected to the second pin of the XOR gate. And the last carry output is the MSB bit of the
output. Worked can be enhanced for the high bit information.

IV. SIMULATION RESULT

VHDL is a standard hardware description language, which offers many useful features for hardware designing. It is a general purpose hardware description language that is easy to use and learn. It allows different levels of abstraction to be mixed in the same model. The level of abstraction used in terms of RTL, Behavioral, switches, gates code. Very popular and industry standard logic synthesis support Verilog HDL, thus designing a chip in Verilog HDL allows the widest choice of vendors. The Programming Language Interface (PLI) is the most powerful feature which allows the programmer to use in

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 7.741ns
```

Fig. 6: Timing Summary of 4-bit BCD Adder using KSA

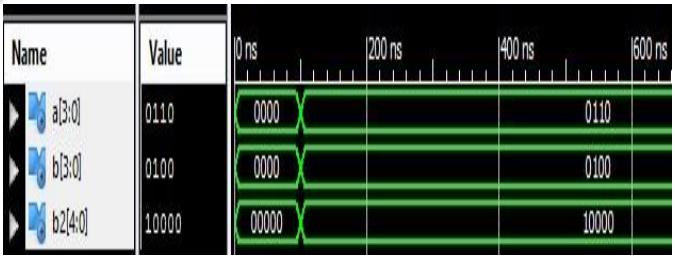
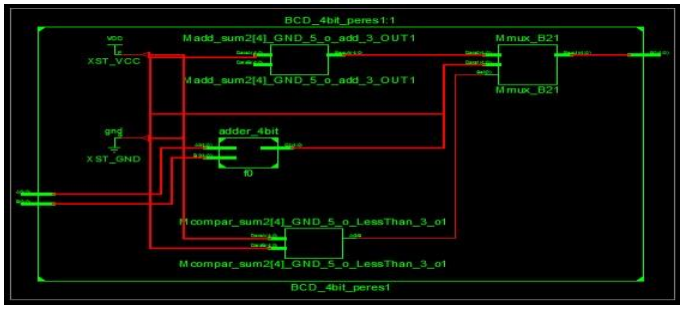


Fig. 7: Output Waveform of 4-bit BCD Adder using KSA

variant way; therefore, a designer can customize the Verilog design simulator as according to their needs by using PLI. View technology schematic of 4-bit reversible adder using Peres gate is shown in fig. 4. In this figure A and B is the two input of the BCD adder with 4-bit word length, B2 is the output of the BCD adder with 5-bit word length.

Fig. 4: View Technology Schematic of 4-bit Adder using KSA



KSA

RTL view of 4-bit adder using KSA is shown in fig. 5. Device ultilization summary of 4-bit BCD adder using KSA is shown in Table II and timing summary of the 4-bit reversible BCD adder using KSA is shown in fig. 6.

Table II: Device Utilization Summary of 4-bit BCD Adder

bcd_64bit_peres Project Status (06/20/2017 - 16:19:20)			
Project File:	BCD_adder_sutractor.xise	Parser Errors:	No Errors
Module Name:	BCD_4bit_peres1	Implementation State:	Synthesized
Target Device:	xc6slx4-3tqg144	• Errors:	No Errors
Product Version:	ISE 14.1	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	9	2400	0%
Number of fully used LUT-FF pairs	0	9	0%
Number of bonded IOBs	13	102	12%

KSA

KSA is a digital circuit it is used four binary multiplier. There are two inputs is word length (7 downto 0) and one output is word length (7 downto 0) shows in Fig. 8. All the input and output of the defined in entity portion and shows the VHDL codes write in begin below. Resister Transfer Level of the Modified Kogge stone adder shows in Fig. 9.

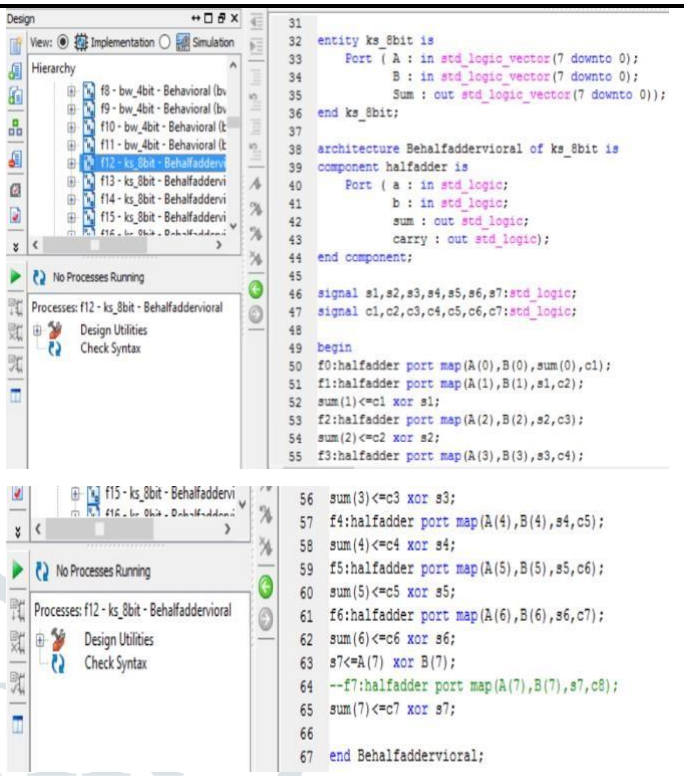


Fig. 8: VHDL Code of Modified Kogge Stone Adder

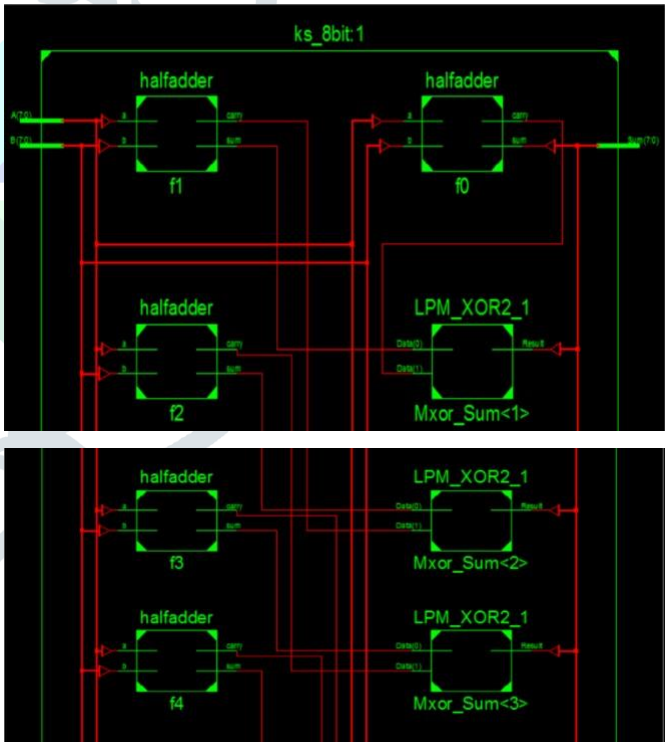


Fig. 9: RTL of Modified Kogge Stone Adder

Output waveform of the Modified Kogge stone adder is shows in Fig. 9. Fig. 10 clearly that the input of the Modified Kogge stone adder is 00110011, 00001111 and output of the Kogge stone adder is 00111010.

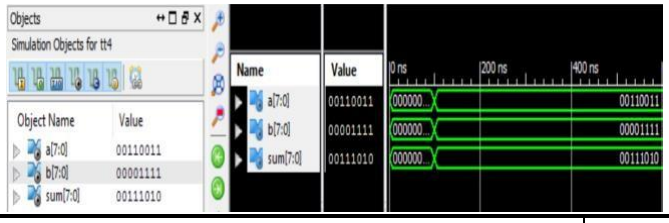


Fig. 10: Output Waveform of Modified Kogge Stone Adder

V. CONCLUSION

Though we focused on BCD adder, we consider the binary adder also because a binary adder is a part of a BCD adder. One can boost the performance of a BCD adder by optimizing the binary adder used in it. Not only in a BCD adder, but a High speed and low power binary adder is also a need in many applications like Filter design, transform.

REFRENCES

[1] Nikhil Advaith Gudala, Trond Ytterdal, John J. Lee and Maher Rizkalla, "Implementation of High Speed and Low Power Carry Select Adder with BEC", IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021.

[2] Krishna Sravani Nandam;K. Jamal;Anil Kumar Budati;Kiran Mannem; Manchalla. O. V. P. Kumar, "Design of Multiplier with Dual Mode Based Approximate Full Adder", 5th International Conference on Communication and Electronics Systems (ICCES), IEEE 2020.

[3] Muteen Munawar;Talha Khan;Muhammad Rehman;Zain Shabbir;Kamran Daniel;Ahmed Sheraz;Muhammad Omer, "Low Power and High Speed Dadda Multiplier using Carry Select Adder with Binary to Excess-1 Converter", International Conference on Emerging Trends in Smart Technologies (ICETST), IEEE 2020.

[4] Maytham Allahi Roodposhti;Mojtaba Valinataj, "A Novel Area-Delay Efficient Carry Select Adder Based on New Addone Circuit", 9th International Conference on Computer and Knowledge Engineering (ICCKE), IEEE 2019.

[5] N. M. Hossain;M. A. Abedin, "Implementation of an XOR Based 16-bit Carry Select Adder for Area, Delay and Power Minimization", International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE 2019.

[6] Deepti Gautam and Anshuman Singh, "Implementation of DHT Algorithm for a VLSI Architecture" Journal of Basic and Applied Engineering Research, Volume 5, Issue 7; OctoberDecember, 2018.

[7] Low Power Approximate Adders for General Computing Using Differential Transmission Gate" on 28.03.2018 National Conference on Innovations in Communication and Computing NCICC" 18 SNS College of Technology.

[8] Omid Akbari, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram, "Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 34, Issue 7, 2017.

[9] G.Challa Ram and D.Sudha Rani, "Area Efficient Modified Vedic Multiplier", 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT).

[10] G. Gokhale and P. D. Bahirgonde, "Design of Vedic Multiplier using Area-Efficient Carry Select Adder", 4th IEEE International Conference on Advancesin Computing, Communications and Informatics (ICACCI-2015), Kochi, August10-13, 2015, India.

[11] G. Gokhale and Mr. S. R. Gokhale, "Design of Area and Delay Efficient Vedic Multiplier Using Carry Select Adder", 4th IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI-2015), Kochi, August 10-13, 2015, India.

[12] Shirali Parsai, Swapnil Jain and Jyoti Dangi, "VHDL Implementation of Discrete Hartley Transform using Urdhwa Multiplier", 2015 IEEE Bombay Section Symposium (IBSS).

[13] Doru Florin Chiper, Senior Member, IEEE, "A Novel VLSI DHT Algorithm for a Highly Modular and Parallel Architecture", IEEE Transactions on Circuits and Systems—Ii: Express Briefs, Vol. 60, NO. 5, May 2013.

[14] Sushma R. Huddar and SudhirRao, Kalpana M., "Novel High Speed Vedi Mathematics Multiplier using Compressors", 9781 - 4673-5090-7/13/$31.00 ©2013 IEEE.