International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 3, May-Jun-2023, ISSN (Online): 2395-566X

Implementation of Multi-Operand B/D Adder for Low Power Dissipation

M. Tech. Scolar Varsha Tumbhurne, Prof. & H.O.D. Dr. Bharti Chourasia
Department of ECE,
RKDF IST,
Bhopal

Abstract- High speed, low power consumption and smaller area are some of the most important criterion for the fabrication of DSP systems and high performance systems. In view of increasing distinction of commercial, economic and internet-based applications that process data in decimal arrangement. In this paper, a new architecture decimal addition of Binary Coded Decimal (BCD) operands, which is the main design part of high speed low power multi-operand binary adders based on this add-3 digit BCD adder, new architectures for higher order (n-digit) BCD adders such as ripple carry adder are developed. The main goal of the proposed algorithm is to perform very much capable fixed bit binary to BCD conversion in terms of delay, power and area. As mentioned earlier, most of the recently proposed adder uses 16-bit binary to BCD converters. The proposed algorithm has been deliberately designed for such converters.

Keywords- BCD, VHDL, Adder, DSP.

I. INTRODUCTION

In virtual world all digital integrated circuit designs, the addition operation is one of the most essential and contains numerous operations. Instruction sets of the DSP's and general principle processors and chips comprise at least one type of addition, subtraction, division, multiplication instruction and other instructions such as subtraction and multiplication employ addition in their operations to perform various operation , and their underlying hardware and software is similar if not identical to addition hardware and software.

Frequently, an adder or multiple adders will be in the critical path of any digital design, hence the performance of a digital design will often be limited to the performance of various adders.

II. ALGORITHM

1. (Shift and Add by constant) of- 7bit:

Fig 1. Shift Add by Constant.

Though the shifting and adding by 3 algorithms is not novel, the architecture execution by means of adding by constant which ultimately makes it area efficient is given away in figure 1. The most important objective of proposed algorithm is to perform proficient fixed bit binary coded decimal conversion. [24]

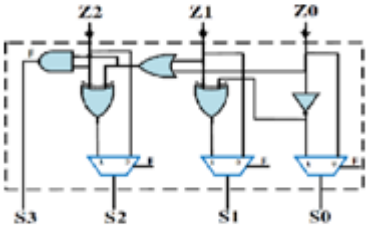

Fig 2. 3 -bit Conditional Adder block.

III. ARCHITECTURAL IMPLEMENTATION

This architecture is different as compared to other architecture. This structure can cover all 82 possible states (0 to 81) and does not need any extra hardware for its implementation which makes the circuit suitable and more dynamic for all applications. This architecture has two different modules. Figure2 and 3 shows the following modules.

- 3 -Bit Conditional Adder Blocks
- 4 -Bit Conditional Adder Blocks.

As shown in figure2 we verify to facilitate if the worth of first 3-bit is greater than (4)10, we will add plus three or else the bit will propagate without any further change as shown in figure 3, we compare to facilitate but the value of the next 4 bit is greater than (4)10 if so, we will add

plus three to the bit or else the bit will be propagated without any further changes in the output.

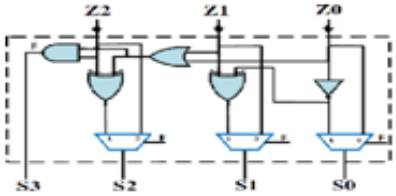

Fig 3. 4 -bit Conditional Adder Block.

IV. METHODOLOGY

The fundamental idea is to shift the binary number left, one bit in one time, into position revert for the BCD results. Let us receive the mode of the binary number 8‟h7C. This is to be modifying into the 12-bit/3 digital BCD result as it's shown below. After the 8 shift operations, the three BCD digits enclose individual: hundredth digit = 4‟b0001, tens digit = 4‟b0010 and ones digit = 4‟b0100, thus representing the BCD assessment of 124.

The idea following the algorithm can be stated as follows:-

- Each time the number is shifted left; it is multiplied by 2 as it is change to the BCD position;
- The value inside the BCD digits is the similar as binary till 9 binary numbers or less than 9 binary numbers. Though if it is 10 or higher than it is not correct BCD number because for BCD, this should carry over after that digit. An improvement has to be needed and this can be made through adding 6 to this binary digit.
- The simplest approach to do this is to distinguish if the value inside the BCD digit locations are 4 or above previous to the shift (i.e. X2). If it is ≥4, then add 3 to the value (i.e. adjust by +6 after the shift). [1]
- The hardware to achieve binary toward BCD conversion is shown below. Shifting is simple – just wire all signals one location to the left. For every one of the BCD locations, we need an "adjust" module which execute and go behind operation: if the value is ≥4, then add 3. This is best illustrated using our example;-

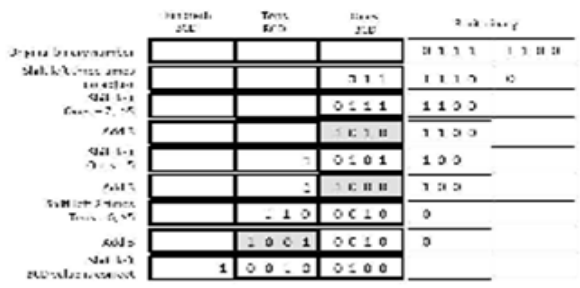

Fig 4. 8-bit Binary to BCD Conversion using Add-3 Algorithm.

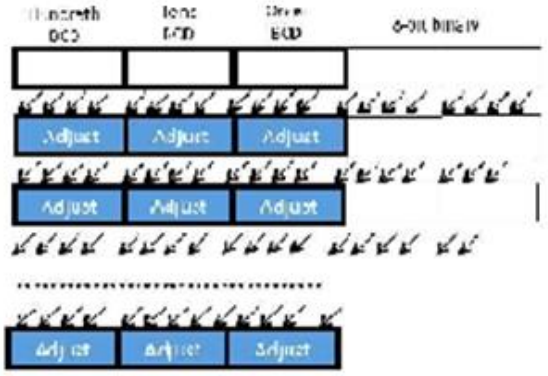

Fig 5. 8-bit Binary to BCD Shifting.

The Binary coded decimal converter works in following steps [1]

- The binary number is shifted by every 1 bit.
- Three columns are segregate hundreds, tens and units o f 4 bit each from left to right, since the binary number is shifted to left individually, after 8 shifting. In the BCD to
- Binary converter module the add-3 component is employed as a component and is given in fig.2 in the block diagram of BCD converter.
- Binary number organized by first is hundreds seconds is tens and third is the units column with most significant bit in hundreds column.
- If the value of binary number in any of the BCD column is four or greater, three is added to so as to BCD column.
- Step 1 is performed once more till each bit of binary number is shifting intended for adding three , a add-3 component is created which maps the charge of column(when greater than 5) to their added result(result after adding three).[1],[30]

V. SOFTWARE PLATFORMS: XILINX ISE WEB PACK 14.1

The Xilinx ISE organization is an incorporate part of design surroundings that consist of a set of plan to generate (capture), suggest and implement digital designs in a field programmable gate array or complex programmable logic design target device. The main purpose is to give an attractive features and simple to use graphics user interface next to online facilitate.

VI. SIMULATION ANALYSIS

Once the all VHDL modules are prepared, they should be simulated before they are put in actual hardware chip. We can generate a test counter waveform from the Project, New Source menu of ISE and it will support in setting up the simulation. Once we simulate our design and feel it is functioning properly, and then we can move on to generating the data needed to essentially program the objective tool with our system design.

VII. ADDER IMPLEMENTATION

Execution of 16 bit Binary to Binary Coded Decimal Converter Adder using add-3 algorithm, addition RCA has been done using Xilinx 14.1 and simulator has commend out by ISim 14.1e tool.

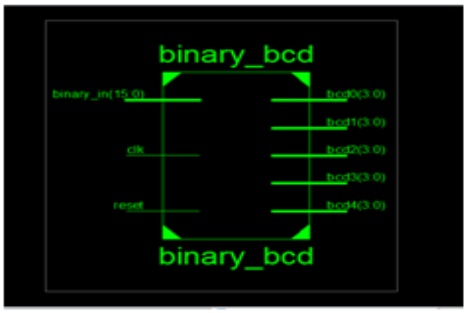

Fig 6. RTL View of 16 bit BCD Converter.

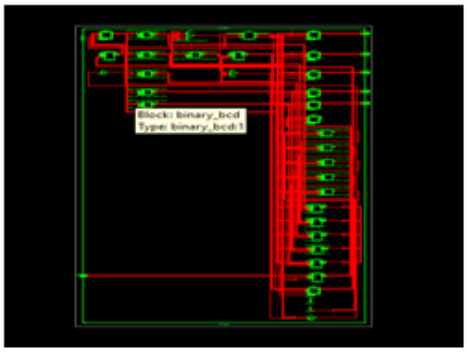

Fig 7. BCD Input Output View RTL View.

Fig 6. Shows resistor transistor logic view of 16 bit Binary to BCD converter in which binary_in, clk,rst are inputs which is going to (N-1 down to 0) bcd0, bcd1, bcd2, bcd3, bcd4: are outputs and figure shows the adder implementation of our proposed work.

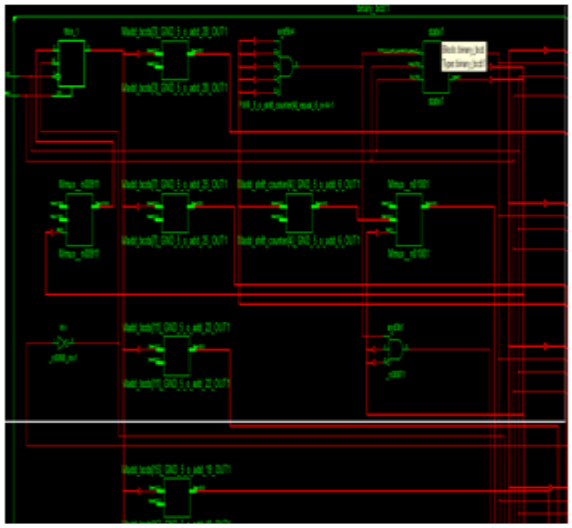

Fig 8. Top Level View of Binary Coded Decimal converter.

Fig 7 represnts the various combination of multiplexers and flip flops and I/O lines related to LUTs and combination of various Binary to BCD adder connected to I/O lines.By double clicking on any of the block gives information about the type of Binary to BCD adder.

Fig.8 shows the Binary Coded Decimal Converter top level view of sequential output like counter and flip flop which counts signal propagation and shows I/O lines of mux and flip flops.

VIII. FPGA ARCHITECTURE

FPGA stand for Field Programmable Gate Arrays .An FPGA is a device that stands on the thousands or even millions of transistors added to perform various logic functions. They perform functions from simple addition and subtraction to multifaceted digital filtering and mistake detection and correction. Aircraft, automobiles, radar, missiles and computers are immediately several of the arrangement that uses FPGA. Figure 9 shows FPGA architecture. I/O edge are the medium in which data are interfaced and sent from internal logic to external sources and from which data are received from external sources. The boundary signals are able to be unidirectional or bidirectional, and single-ended or differential

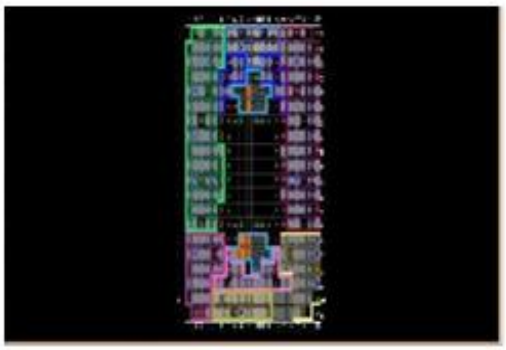

Fig 9. FPGA CLB Blocks.

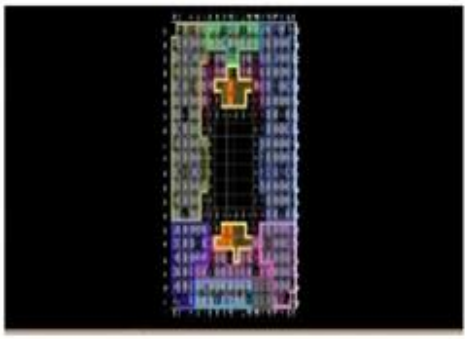

Fig 10. FPGA Floor Planer.

By means of the FPGA editor and Floor planner of the ISE tool the organization can be end more efficiently in word of space acquired when assembled on a real hardware chip shown in fig 10.

IX. FPGA DESIGN FLOW

The FPGA architecture develops and its difficulty increases, CAD software has become more grown-up as well. Today, the majority of the FPGA retailer affords a fairly complete set of intend tools that allows routine synthesis and compilation from design of the specifications in hardware specification languages, such as Verilog or VHDL, all the way behind to a bit flow to program FPGA chips. Xilinx‟s basic structure blocks are known as (CLBs). Each CLB contains piece, and each and every slice has (LUTs).

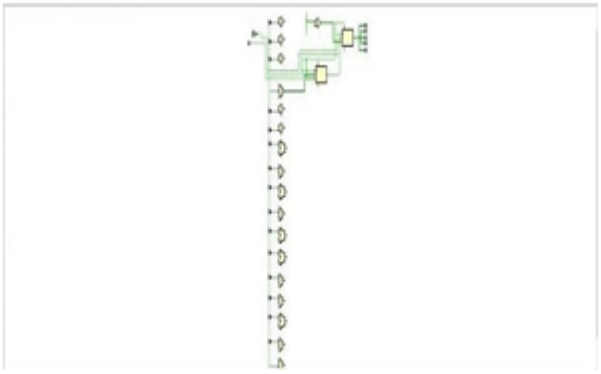

Fig 11. FPGA Logic.

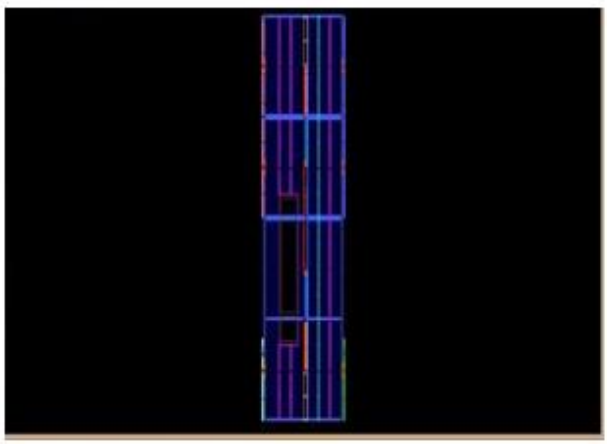

Fig 12. FPGA Block.

.

X. FPGA PROGRAMMING

A programming file is created by running the Generate Programming File process. This process is being run subsequent to the FPGA intend has been totally routed. The Generate Programming File procedure simulates Bit Gen, the Xilinx bit stream generation program, to generate a bit stream (.BIT) or (.ISC) file for Xilinx device configuration. The FPGA tool is then configured by means of the bit file using the JTAG edge scan method. After the Spartan device is configured for the intended of the design, then its working is verified by applying different type‟s inputs.

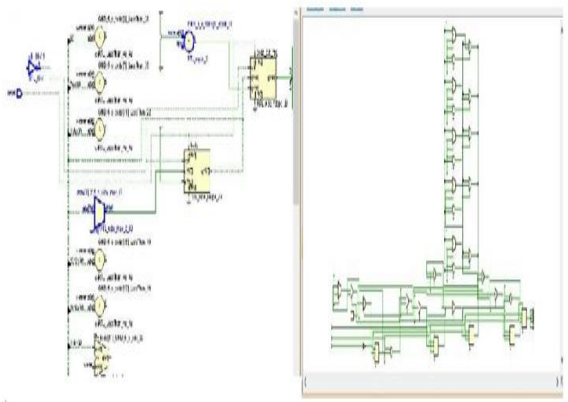

Fig 13. FPGA Logic.

XI. SIMULATION RESULTS

Here below shows the result analysis of our proposed work. Proposed work is being compared with the existing design. According to the existing design they designed 7 bit Binary to BCD converter through add 3 conversion algorithms and synthesized in synopsis tool. The proposed thesis in which a 16 bit Binary to BCD converter is designed through add 3 algorithm which is Binary to BCD converter as well as adder in which shifting and adding process takes place. Adding is done when digit is greater than 4 so add the digit by 3. The proposed work shows superior results as compared to existing design. Table 1 shows the comparison of Binary Coded Decimal Converter with existing design [1]. Binary Coded Decimal adder is a modest additional but it provides option for addition with large number of input and also in more efficient way in expression of area and delay.

Table 1. Comparison Table.

Metric	Area (µm2)	Delay(ns)	Power(w)
Proposed design	48 LUT used out of63,400 Utilization 1%	1.663	0.42
Design [1]	903	1.89	549

Table1 below shows the comparison of Binary Coded Decimal Converter with existing design [1]. Synthesis results demonstrate that present is a reduction in delay, power and area. This in turn reduces power delay product and Minimum period: 2.600ns (Maximum Frequency: 384.645MHz) & Minimum input arrival time before clock: 2.765ns & Maximum output required time after clock: 4.714ns Maximum combinational path delay: Fig.14 shows the timing summary in which timing details, delay, data path and total delay of the circuit of 16-bit Binary Coded Decimal Converter.

Simulation process and Results: In this work, Xilinx and I-sim tools are used for timing analysis and synthesis. The simulation output for both 16-bit Binary to BCD adder is presented. After verifying the block diagram, the behavior

of 16-bit Binary to BCD adder is checked by simulation process. Fig. 15 shows the simulation waveform of Binary Coded Decimal Converter in which state and state _next signal waveform of Binary Coded Decimal Converter shown. Fig.16 shows the simulation waveform of Binary to BCD converter which represents Binary Coded Decimal Converter simulation output, which shows reset on 0 and Binary Coded Decimal out register occupied output value and remaining length value represented by "L"

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

   Minimum period: 1.662ns (Maximum Frequency: 601.273MHz)
   Minimum input arrival time before clock: 0.789ns
   Maximum output required time after clock: 0.643ns
   Maximum combinational path delay: No path found

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 1.662ns (frequency: 601.273MHz)
  Total number of paths / destination ports: 451 / 124
-------------------------------------------------------------------------
Delay:               1.662ns (Levels of Logic = 2)
  Source:            shift_counter_4 (FF)
  Destination:       binary_0 (FF)
  Source Clock:      clk falling
  Destination Clock: clk falling

  Data Path: shift_counter_4 to binary_0
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDCE:C->Q             4   0.361   0.697  shift_counter_4 (shift_counter_4)
     LUT5:I0->O           37   0.097   0.403  PWR_5_o_shift_counter[4]_equal_6_o<4>1 (PWR_5_o_shift_counter[4]_equal_6_o)
     LUT3:I2->O            1   0.097   0.000  state_FSM_FFd2-In1 (state_FSM_FFd2-In)
     FDC_1:D                   0.008          state_FSM_FFd2
    ----------------------------------------
    Total                      1.662ns (0.563ns logic, 1.100ns route)
                                       (33.9% logic, 66.1% route)
```

Fig 14. Timing Summary of 16- bit Binary Coded Decimal Converter.

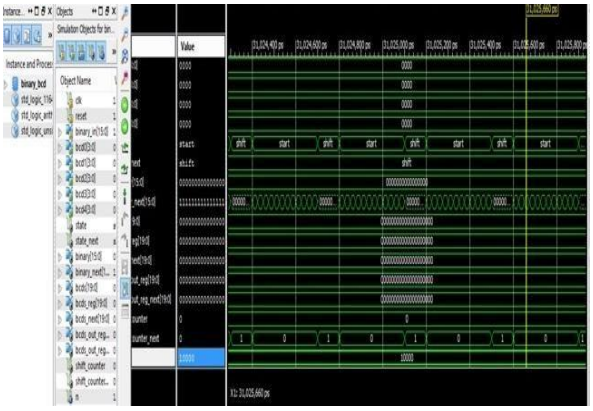

Fig 15. Simulation Waveform of Binary Coded Decimal Converter.

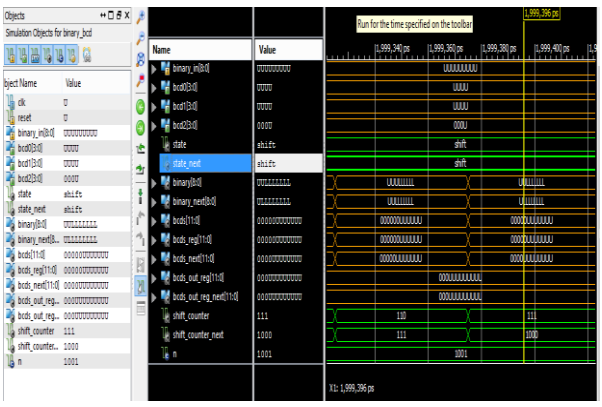

Fig 16. Simulation Waveform of Binary Coded Decimal Converter.

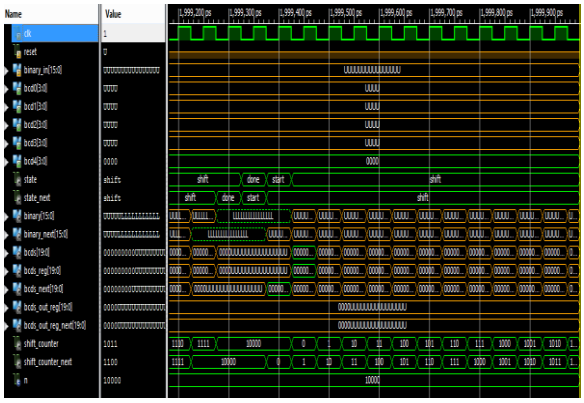

Fig 17. Simulation Waveform of Binary Coded Decimal Converter.

Fig 17 represents Binary Coded Decimal Converter simulation output, which shows reset on „U" and CLK on „1" that means clock is applied here in this part and the output values are changed according to that in which the number and its BCD number is converted simultaneously and counter also shown which counts the value till 16. Here start, shift and done shows the conversion and addition process of our proposed work.

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	63	126,800	1%	
Number used as Flip Flops	63			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	48	63,400	1%	
Number used as logic	48	63,400	1%	
Number using O6 output only	47			
Number using O5 output only	0			
Number using O5 and O6	1			
Number used as ROM	0			
Number used as Memory	0	19,000	0%	
Number used exclusively as route-thrus	0			
Number of occupied Slices	20	15,850	1%	
Number of LUT Flip Flop pairs used	68			
Number with an unused Flip Flop	6	68	8%	
Number with an unused LUT	20	68	29%	
Number of fully used LUT-FF pairs	42	68	61%	
Number of unique control sets	3			

Fig 18. Summary of Binary Coded Decimal Converter.

XII. POWER ANALYSIS

Binary coded decimal adder is a modest additional but it provides option for addition with large number of input and also in more efficient way in expression of area and delay. So proposed work is very efficient and useful in higher level addition with increasing numeral of inputs with the increasing complexity in computation application. Power analysis figure of the proposed design shows the total power consumed by the design in terms of leakage current and also tell about the thermal properties.and supply power of the proposed design. Figure below shows all the output parameters related to power analysis of the proposed design is shown the fig 19. All these parameters of the proposed design show superior results in terms of delay, power, and area. The proposed design is simulated and synthesized in Xilinx 14.1.

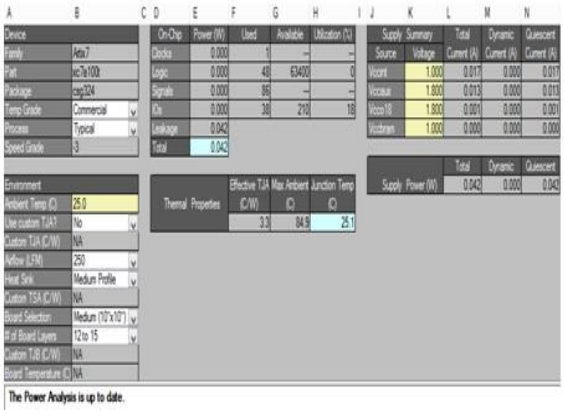

Fig 19. Shown Power expenditure of Binary Coded Decimal Adder.

XIII. CONCLUSIONS

A novel integrated Binary to Binary Coded Decimal multi-operand addition algorithm has been proposed. The binary parallel multi-operand addition is realized by programmers to convert a Binary number to Decimal. It is performed by shifting and adding through add-3 algorithm [1], and can be implemented using a smaller amount of number of gates in computer hardware, which ultimately makes it area efficient.

The proposed Binary Coded Decimal converter forms the center of the multi-operand Binary adder. Simulation results show the competence of our proposed BCD converter in addition to multi-operand decimal adder with respect to exiting designs. [1]

REFERENCES

[1] Ashish Joshi, Tooraj Nikoubin,Area-Efficient and Power-Efficient Binary to BCD Converters Sri Rathan Rangisetti, Member, IEEE Dept. of Electrical & Computer Engineering, Texas Tech University, Lubbock, TX, 79409 2020

[2] J. Ditmar and S. Mc Keever Array Synthesis in System C Hardware Compilation 2021

[3] H. Neto and M. Vestia Decimal Multiplier on FPGA using Embedded Binary Multipliers 2018

[4] S. Jang. et al Cache Miss-Aware Dynamic Stack Allocation.2022

[5] Praveena Murugesan Ranganathan Engineering College, Coimbator Electronic Engineering

[6] Emre Salman Design of Efficient Reversible BCD Adder–Subtractor Architecture and Its Optimization Using Carry Skip Logic:.2001

[7] Ashis Kumer Biswas, Md. Mahmudul Hasan, Ahsan Raja Chowdhury, Hafiz Md. Hasan BabuMJ Efficient approaches for designing reversible Binary Coded Decimal adders 2008

[8] AK Biswas A novel approach to design BCD adder and carry skip BCD adder 2001

[9] Hafiz Md Hasan Babu, Md Rafiqul Islam, Ahsan Raja Chowdhury, Syed Mostahed Ali Chowdhury

[10] Variable block carry skip logic using reversible gates M Islam 2002.Reversible logic synthesis for minimization of full-adder circuit2003

[11] generation Bayrakci and AhmetAkkas BCD adder with efficient carry (2007)

[12] AnshulSingh A novel architecture for BCD addition and subtraction (2009)

[13] unified architecture for BCD and binary addition ChetanKumar (2011)

[14] Sundaresan (2011)a correction free BCD , Al-khaleel (2011).a pioneer work on design of Reduced delay BCD adder using Carry Look Ahead(CLA

[15] Technology C. Sundaresan Design and Synthesis of Reduced Delay BCD International Journal of Computer and Information (ISSN: 2279 – 0764) Volume 04 – Issue 01, January 2015

[16] Optimized design of BCD adder and Carry skip BCD adder using reversible logic gates H R Bhagyalakshmi et al. / International Journal on Computer Science and Engineering (IJCSE)

[17] MILOS˘ D. ERCEGOVAC and T. Lang, Digital Computer Arithmetic. Elsevier/Morgan Kaufmann Publishers, 2004.

[18] A Decimal / Binary Multi-operand Adder using a Fast Binary to Decimal Convert, Rishabh Panday 2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems

[19] T.J Sunil Daya Sagar MAR, M. J. Schulte. High-speed multi-operand decimal adders. IEEE Trans. on Computers, 54(8):953–963, Aug.2005.

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

[21] Decimal Floating-Point Multiplication Via Carry-Save Addition Mark A. Erle International Business Machines 6677 Sauterne Drive Macungie, PA 18062 merle@us.ibm.com Michael J. Schulte and Brian J. Hickmann University of Wisconsin – Madison Dept. of Electrical and Computer Engineering Madison, WI 53706 schulte@engr.wisc.edu and bjhickmann @wisc.edu

[22] Novel BCD Adders and Their Reversible Logic Implementation for IEEE 754r Format Himanshu Thapliyal, Saurabh Kotiyal and M.B Srinivas Center for VLSI and Embedded System Technologies, International Institute of Information Technology, Hyderabad-500019, India *Department of Computer Engineering, SIT, Kukas, Jaipur, India (thapliyalh imanshu@yahoo.com, saurabhkotiyal@yahoo.com, srinivas@iiit.net)

[23] Jeff Rebacz, Erdal Oruklu, Jafar Saniie Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA E-mail: erdal@ece.iit.edu Received April 14, 2011; revised May 22, 2011; accepted May 29, 2011

[24] Decimal Floating-Point: Algorism for Computers Michael F. Cowlishaw IBM UK Ltd., P.O. Box 31, Birmingham Road, Warwick CV34 5JL, UK or Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK mfc@uk.ibm.com
[25] 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.
[26] 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.
[27] Ranaganathan Panchagam "Minimization of Power Dissipation in Digital Circuits Using Pipelining and a Study of Clock Gating Technique" Master‟s Thesis, Conference (NEWCAS), 2012 IEEE 10th International. IEEE, 2012
[28] Dadda, Luigi. "Multi-operand parallel decimal adder: A mixed binary and bcd approach." Computers, IEEE Transactions on 56.10 (2007): 1320-1328
[29] 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.
[30] Michael Keating, David Flynn, Robert Aitken, Alan Gibbons, Kaijian Shi "Low Power Methodology Manual for System-On-Chip Design" , Springer, 2007
[31] araju P. Mohanty, Nagarajan Ranaganathan, Elias Kougianos, Priyadarsan Patra "Low-Power High-Level Synthesis for Nan scale CMOS Circuits", Springer, 2008
[32] Vikas Kumar, Cadence Design Systems, Inc. "Low-Power CMOS Circuit Design" in http://www.powe rmanagementdesignline.com/howto/189500236
[33] Saeeid Tahmasbi Oskuii "Design of Low-Power Reduction-Trees in Parallel Multipliers" Ph.D. Dissertation, Norwegian University of Science and Technology, 2008.
[34] Sataporn Pornpromlikit "Power-Efficient Design of 16-Bit Mixed Operand Multipliers" Master‟s Thesis, Massachusetts Institu te of Technology, 2004.