iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** V **Month of publication:** May 2020

DOI: http://doi.org/10.22214/ijraset.2020.5151

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

Implementation of 4x4 Data bit Multiplier using Dadda Algorithm and Optimized Full Adder

Siddhardha Sathanapally[1], K. Sai Sreeja[2], S. K. Ahmed Baig[3], Dr. Syed Jahangir Badashah[4]
[1, 2, 3]*Sreenidhi Institute of Science and Technology, Ghatkesar.*
[4]*Associate Professor, Department of Electronics and Communications Engineering, Sreenidhi Institute of Science and Technology, Ghatkesar.*

Abstract: Multiplier factor is one in all the foremost necessary arithmetic modules within the quick computing applications. Multipliers and their associated circuits like [*fr1] adders, full adders and accumulators consume a major portion of most high speed applications. so as to scale back the hardware that ultimately reduces a vicinity, power and propagation delay, economical full adders are utilized in the multipliers. They additionally scale back the ability and propagation delay of the multiplier factor. during this paper 4x4 multiplier factor is intended for mistreatment. Dadda algorithmic rule and 10T full adder. Use of Dadda algorithmic rule reduces ten.47% of the ability dissipation, 27.67% of the propagation delay and thirty five.34% power delay product compared to Array multiplier factor. The simulations ar performed mistreatment Tanner EDA of 45nm technology
Keywords: Dadda multiplier, VHDL, Verilog, Luigi Dadda, Amorphous Algorithm

I. INTRODUCTION

Augmentation is that essential procedure that's utilized in varied electronic and in varied advanced correspondence applications. Multipliers with low idleness and least force dissemination unit likeable to rearrange an economical circuit therefore greatest turnout could also be accomplished in the least latent period. Building squares utilized in multipliers unit a full diabolical and a half multitude. Distinctive preparatory executions of a full diabolical associate degreed half multitude circuits are wont to decrease force associate degreed deferral in morder to structure an upgraded range circuit that contains pass junction transistor clarification, CMOS methodology innovation, split rate knowledge driven clarification and CMOS methodology innovation. Other than this, distinctive augmentation calculations likewise are wont to accomplish accrued force and hold over items that contain Dadda, Wallace tree, and sacred writing and Booth calculations.

II. EXISTING TECHNOLOGY

It is the traditional multiplier factor. It uses similar steps as utilized in the traditional multiplication. it's supports shift and add algorithms. 4x4 multiplier factor uses sixteen AND gates, four [*fr1] adders, eight full adders and twelve total adders are used [2]. In Fig.1 a0, a1, a2 and a3 are bits within the number and b0, b1, b2 and b3 are the bits in multiplier factor. P0, p1, p2, p3, p4, p5, p6 and p7 are the output bits. Where angular distance is that the [*fr1] adder and solfa syllable is that the full adder.

III.PROPOSED IDEA

Array multiplier factor has a lot of power consumption and propagation delay compared to alternative multipliers. Wallace multiplier factor has giant space wastage drawback thanks to irregularity within the structure. the religious text multiplier factor system becomes complicated for complicated numbers. to beat these disadvantages within the existing multiplier factors Dadda multiplier is projected. Dadda multiplier factor could be a quick parallel multiplier factor bestowed by Luigi Dadda in 1965. it's a refinement of Wallace multiplier factor therefore its algorithmic rule additionally has 3 general stages. The procedure of the 3 stages is the same for Dadda multiplier factor as in Wallace multiplier factor.But in Dadda multiplier factor, the row reduction is processed by putting adders at most heights of the matrix in optimum manner. Thus it requires less adders compared to Wallace multiplier factor. Reduction method is controlled by dj price. wherever dj is that the most height of the sequence, dj=2 then dj+1=floor(15dj)
The sequence is d1=2, d2=3, d3=4, d4=6, d5=9, d6=13….The initial price of dj is chosen because the largest price such dj Where m and n ar the input bits within the input number and multiplier factor. If height (Ci)≤dj then that column doesn't need reduction, move to the next column. If height of of Ci=dj+1 add high 2 components during a [*fr1] adder, place the result at the bottom of the column and carry at the highest of column Ci+1, then move to column Ci+1.Else add high 3 elements during a full adder, place the result at very cheap of the column and carry at the highest of the column Ci+1, restart Ci at step1. 4x4 bit Dadda multiplier factor algorithmic rule method is shown in Fig.4. the amount of full and [*fr1] adders required for the Dadda multiplier depend upon size of the operands, determined by the subsequent equations [5].

Thus the Dadda multiplier factor needs less range of adders compared to all or any alternative multipliers. Therefore space of multiplier factor is reduced .The power dissipation is reduced and speed is additionally reduced, attributable to reduction of partial product that is important at .

IV.RESULTS

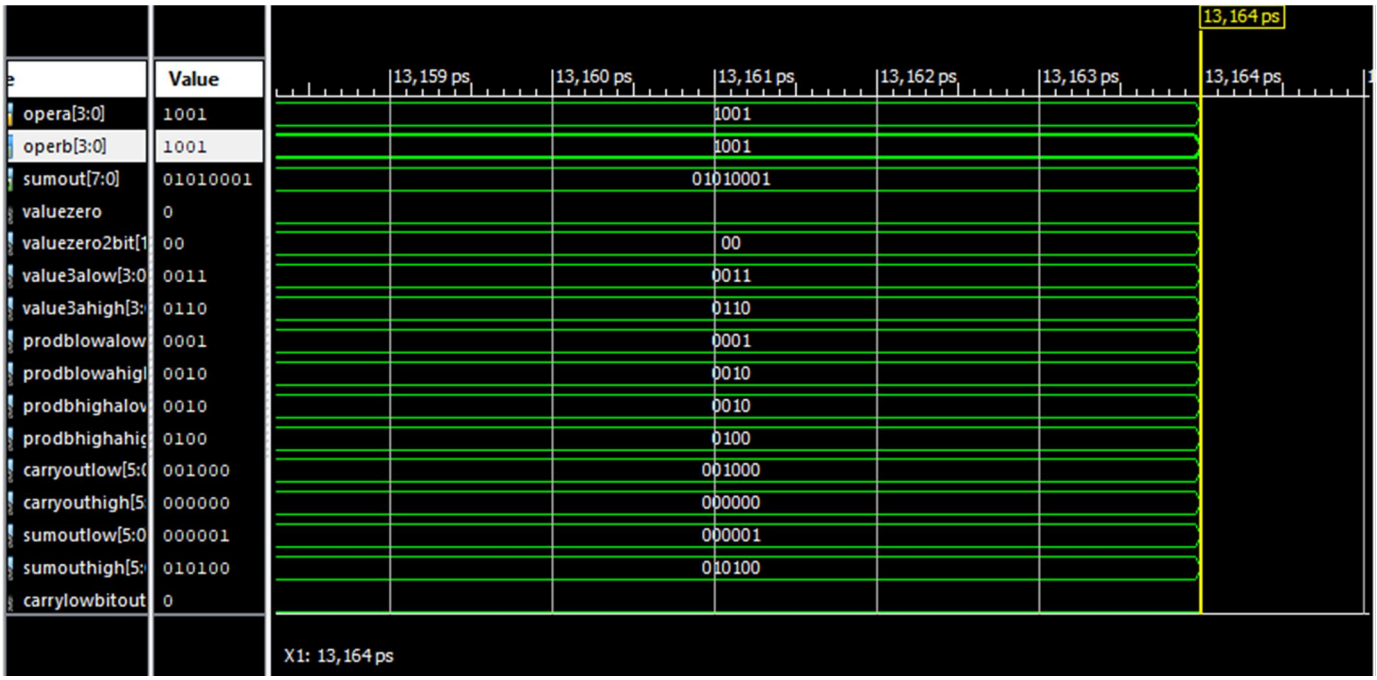


Fig 1. output simulations

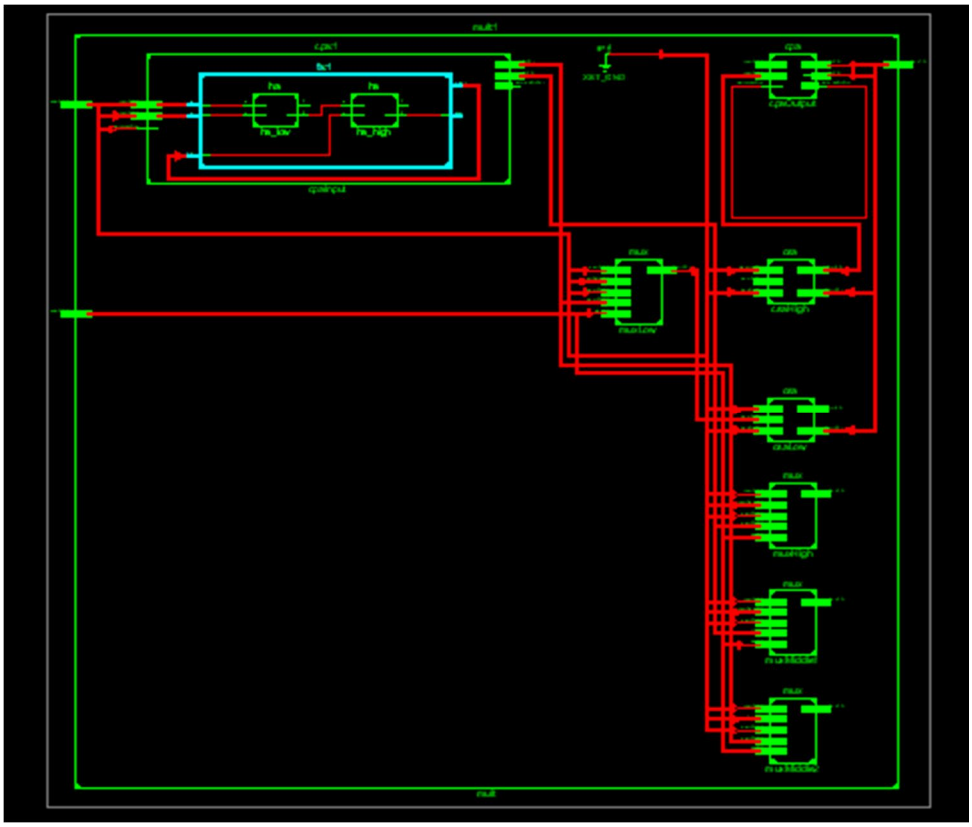


Fig 2. Circuit diagram

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

```
 1
 2
 3  library IEEE;
 4  use IEEE.STD_LOGIC_1164.ALL;
 5  use IEEE.STD_LOGIC_ARITH.ALL;
 6  use IEEE.STD_LOGIC_UNSIGNED.ALL;
 7
 8  entity mult is
 9     port (operA       : in  std_logic_vector(3 downto 0);
10           operB       : in  std_logic_vector(3 downto 0);
11           sumOut      : out std_logic_vector(7 downto 0)
12          );
13  end mult;
14  -- ================================================================================
15  -- ================================================================================
16  architecture rtl of mult is
17
18  constant  zero          : std_logic:='0';
19  signal  valueZero     : std_logic:=zero;
20
21  constant zero2Bit       : std_logic_vector(1 downto 0):="00";
22  signal   valueZero2Bit  : std_logic_vector(1 downto 0):= zero2Bit;
23
24          --  Two below used to transfer result of 3*operA
25  signal  value3ALow    : std_logic_vector(3 downto 0);
26  signal  value3AHigh   : std_logic_vector(3 downto 0);
27
28  signal  prodBLowALow    : std_logic_vector(3 downto 0);
```

Fig 3. Code snapshots

```
Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Logic Utilization:
  Number of 4 input LUTs:                32 out of   1,920    1%
Logic Distribution:
  Number of occupied Slices:             17 out of     960    1%
    Number of Slices containing only related logic:     17 out of      17 100%
    Number of Slices containing unrelated logic:         0 out of      17   0%
      *See NOTES below for an explanation of the effects of unrelated logic.
  Total Number of 4 input LUTs:          32 out of   1,920    1%
  Number of bonded IOBs:                 16 out of      83   19%
```

Fig 4. Synthesis Report

V. CONCLUSION AND FUTURE SCOPE

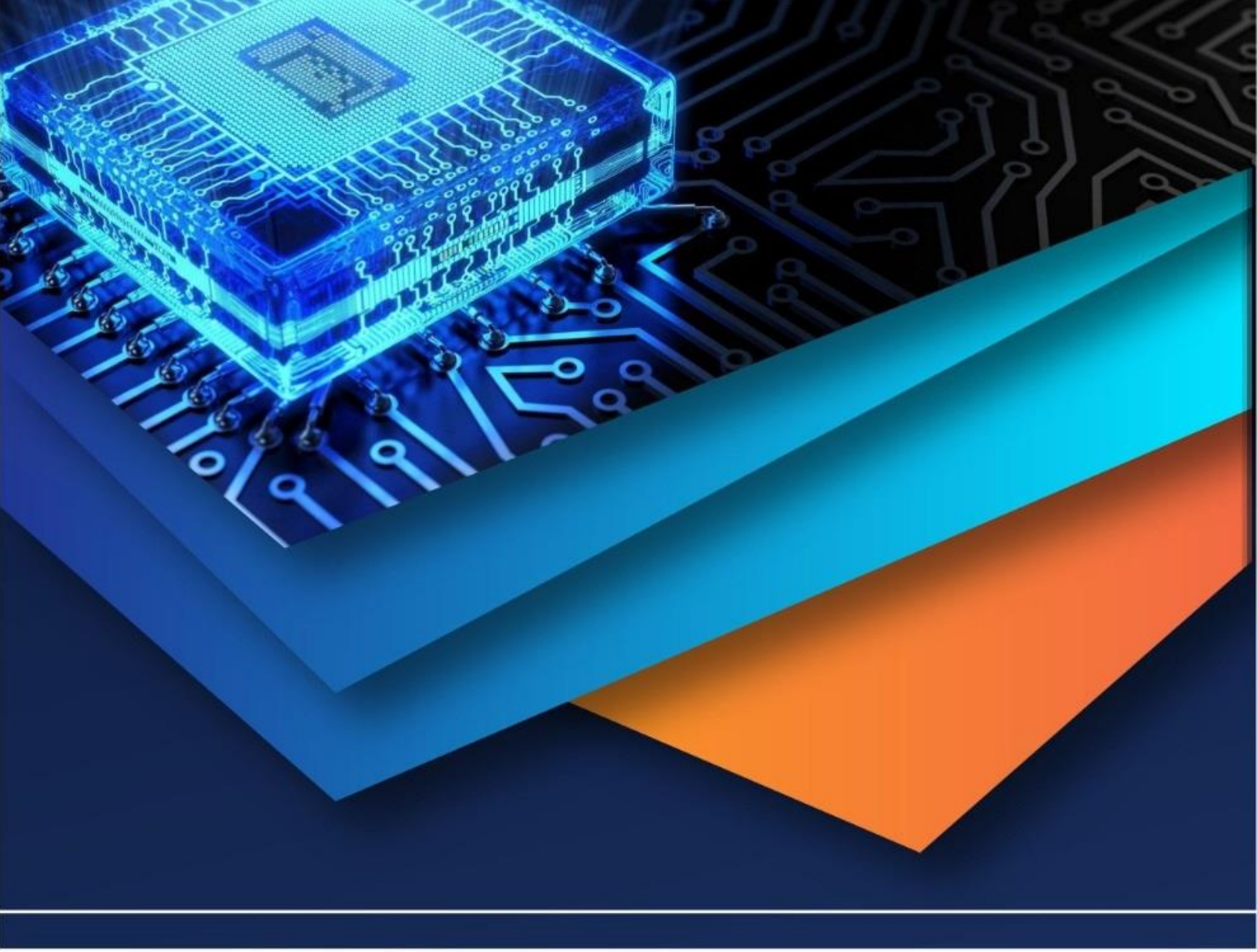
Proposed Dadda multiplier has low power consumption and low propagation delay compared to existing multipliers whereas the Array multiplier has more power consumption, propagation delay and power delay product. Dadda multiplier has achieved 10.47% reduction in power dissipation, 27.67% reduction in propagation delay and 35.34% power delay product compared to Array multiplier. These multiplier designs can be implemented for 8x8 multipliers. Every multiplier has its own advantages and disadvantages using hybrid techniques new multiplier design can be implemented. Dadda multiplier has the complex carry propagate adder at the final stage whereas the Wallace multiplier has less complex carry propagate adder. By combining these two algorithms a new hybrid multiplier can be designed.

VI. ACKNOWLEDGMENT

We wish to express my propound sense of gratitude to Mr. Syed Jahangir Baadshaah ,Associate Professor, Department of ECE, SNIST for his valuable guidance and wonderful insights. We also extend my thanks to Mr. S.P.V Subbarao, Head of Department, Department of ECE, SNIST for extending his help. We express my gratitude to Mr. Syed Jahangir, Associate Professor, Department of ECE, SNIST, for his guidance throughout the project. Finally I express my sincere gratitude to all the members of faculty and my friends who contributed their valuable advice and helped to complete the project successfully.

REFERENCES

[1] Zain Shabbir, Anas Razzaq Ghumman, Shabbir Majeed Chaudhry, A Reduced-sp-D3Lsum Adder-Based High Frequency 4 × 4 Bit Multiplier Using Dadda Algorithm, Springer Science+Business Media New York 2015.
[2] Design of high-speed carry saves adder using carry lookahead adder. Available from: https://www.researchgate.net/publication/301407573_Design_of_high_speed_carry_save_adder_using_carry_lookahead_adder [accessed Sep 22, 2017].
[3] S. Z. Naqvi, S. Z. Hassan and T. Kamal, "A power consumption and area improved design of IIR decimation filters via MDT," 2016 International Conference on Intelligent Systems Engineering (ICISE), Islamabad, 2016, pp. 146-151. doi: 10.1109/INTELSE.2016.7475111
[4] A. Mukhtar, H. Jamal and U. Farooq, "An area efficient interpolation filter for digital audio applications," in IEEE Transactions on Consumer Electronics, vol. 55, no. 2, pp. 768-772, May 2009. doi: 10.1109/TCE.2009.5174452
[5] Stephen P. Boyd, Seung-Jean Kim, Dinesh D. Patil, Mark A. Horowitz, Digital Circuit Optimization via Geometric Programming, Operations Research, v.53 n.6, p.899-932, November-December 2005 [doi>10.1287/opre.1050.0254]
[6] P. Prem Kumar, K. Duraiswamy, and A. Jose Anand, “An optimized device sizing of analog circuits using genetic algorithm,”European Journal of Scientific Research,vol.69,no.3,pp.441–448, 2012.
[7] Revna Acar Vural, Member, IEEE, Burcu Erkmen, Member, IEEE, Ufuk Bozkurt, Tulay Yildirim, Member, IEEE, “CMOS Differential Amplifier Area Optimization with Evolutionary Algorithms, ” Proceedings of the World Congress on Engineering and Computer Science 2013 Vol II WCECS 2013, 23-25 October, 2013, San Francisco, USA.
[8] Jorge Juan Chico, Enrico Macii, book chapter 17, “Power- Consumption Reduction in Asynchronous Circuits Using Delay Path Unequalization, page (151-160)”, 13th International Workshop, PATMOS 2003 Turin, Italy, September 10-12, 2003, Proceedings.
[9] Raminder Preet Pal Singh, Parveen Kumar, Balwinder Singh, “Performance Analysis of 32 bit Array Multiplier with Carrying Save Adder and with Carry Look ahead adder”, in International Journal of Recent Trends in Engineering, Vol 2, No. 6, November 2009
[10] B. Eghbalkhah, B. Afzal, and A. Afzali-Kusha, "Speed improvement algorithm for 16×16 multipliers using sizing optimization," 2007 International Conference on Design & Technology of Integrated Systems in Nanoscale Era, Rabat, 2007, pp. 98-101. doi: 10.1109/DTIS.2007.4449500
[11] Prashant D. Pawale, Venkat N Ghodke, High-speed Vedic multiplier design and implementation on FPGA, IJAR 2015(Pune, India).
[12] M. Aguirre-Hernandez, M. Linares-Aranda, CMOS full-adders for energy-efficient arithmetic applications.IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 19(4), 718–721 (2011).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY