International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5, June 2019

Design & Implementation of Wallace Tree Multiplier and Kogge Stone Adder

Nikita Gulliya, Anjali Baliyan, Geetanjali Raj

Abstract: ***An increment in demand of portable devices makes low power device design and becomes an important field of research. Delay is one of the main design objective in integrated circuits after power. Three parameters which are power, area, and delay are always trade off. However speed and area are usually conflicting constraints, so that speed improvement results mostly in larger area. The multiplication and addition of two binary numbers are the most frequently and fundamental used arithmetic operations in microprocessors, data-processing application specific integrated circuits and digital processors. This research work will focuses on analyzing the delay of a Wallace tree multiplier and a Kogge Stone Adder by optimizing their circuits to increase the speed of the different digital circuits which are used in computations. Multipliers and adders are most important part in the digital processing or other applications. Therefore, multipliers and adders should be designed in such a way that their speed would be high and delay would be low. There are many attempts have been made to reduce the partial products which are generated in the multiplications and by using booth algorithm, whereas Wallace tree carry save adder structure have been used to sum the partial products to reduce time. In the full paper, the full implementation and simulation results will be discussed. Greatresults have beenachieved and comparison study has also been done with previous research work***

Keywords: Wallace tree multiplier, kogge stone adder,

I. INTRODUCTION

An increment in demand of portable devices makes low power device design and becomes an important field of research. Delay is one of the main design objective in integrated circuits after power. Three parameters which are power, area, and delay are always trade off. However speed and area are usually conflicting constraints, so that speed improvement results mostly in larger area. The multiplication and addition of two binary numbers are the most frequently and fundamental used arithmetic operations in microprocessors, data-processing application specific integrated circuits and digital processors. This research work will focuses on analyzing the delay of a Wallace tree multiplier and a Kogge Stone Adder by optimizing their circuits to increase the speed of the different digital circuits which are used in computations. Multipliers and adders are most important part in the digital processing or other applications. Therefore, multipliers and adders should be designed in such a way that their speed would be high and delay would be low. There are many attempts have been made to reduce the partial products which are generated in the multiplications and by using booth algorithm, whereas Wallace tree carry save adder structure have been used to sum the partial products to reduce time.

Manuscript published on 30 June 2019.
***** Correspondence Author (s)
Nikita Gulliya, Agnisys Technology, RnD Engineer, ,India.
Anjali Baliyan, Dept. of Electronics and communication, Maharaja Surajmal Institute of Technology, Delhi, India
Geetanjali Raj, Dept. of Electronics and communication Engineering, ABES Engineering College, Ghaziabad, India

II. IMPLEMENTATION & SIMULATION

A. *Wallace tree multiplier:*

The conventional multiplier is designed first for reference. To find the delay for the conventional Wallace tree multiplier first so that the results could later be compared with the fast multiplier which is proposed with the help of kogge stone adder. Therefore, in the architecture of conventional Wallace tree multiplier 4-bit input A and B are taken so that we get 8-bit output P which is the product for both A and B inputs. Different wires used in between are declared next. The code is written in behavioral style modeling in Verilog coding language. The first step is to calculate the partial products. The partial products are taken my multiplying each bit of A with B, we use AND operation for the same and store the result in separate registers which are used for the addition at a later stage. A being the multiplier and B is the multiplicand. Once the partial products are obtained by the multiplication. We go to the first stage of the addition which uses the 2 half adders and 2 full adders and calls the half adder and full adder modules discussed earlier.

The sum and the carry is then propagated ahead to the second stage which consists of 1 half adder and 4 full adders. Its result further moves down to the third stage with 5 half adder stages to give the final sum to the product of the multiplication. The module was successfully simulated, and output waveforms were obtained and verified with the test bench.

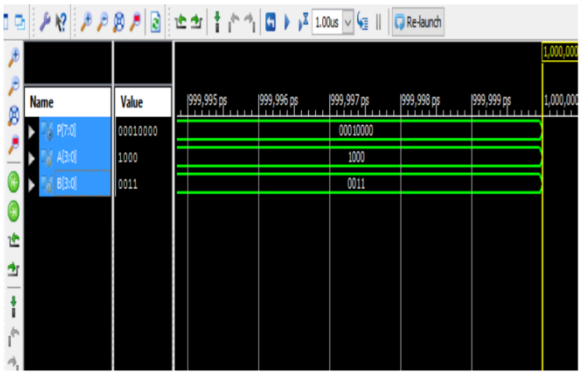

Fig 1: Simulation waveforms for Wallace tree multiplier

```
Timing constraint: Default path analysis
  Total number of paths / destination ports: 432 / 8
--------------------------------------------------------------------------------
Delay:               13.213ns (Levels of Logic = 8)
  Source:            A<0> (PAD)
  Destination:       P<6> (PAD)

  Data Path: A<0> to P<6>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             6   1.218   0.844  A_0_IBUF (A_0_IBUF)
     LUT2:I0->O            3   0.704   0.706  pp2_0_and00001 (pp2<0>)
     LUT4:I0->O            3   0.704   0.535  fa12/carry1 (c12)
     LUT4:I3->O            3   0.704   0.706  fa22/carry1 (c23)
     LUT3:I0->O            4   0.704   0.666  fa23/carry1 (c24)
     LUT4:I1->O            2   0.704   0.622  ha33/Mxor_s_Result1 (c35)
     LUT3:I0->O            1   0.704   0.420  ha34/c1 (P_6_OBUF)
     OBUF:I->O                 3.272          P_6_OBUF (P<6>)
    ----------------------------------------
    Total                     13.213ns (8.714ns logic, 4.499ns route)
                                       (66.0% logic, 34.0% route)

--------------------------------------------------------------------------------
```

Fig 2: Timing delay calculation of Wallace tree multiplier

B. Kogge Stone adder

Black cell: Kogge stone adder has been designed by dividing into sub modules. One such module is the black cell. It is a very important unit of the kogge stone adder design. The inputs which we have taken in it are of 1-bit each and there are 4 inputs in total. The generation and propagation of carry is taking place in it with the help of computation using AND and OR gates. The 4 inputs are G (generation) and P (propagation) of present state and previous stage carry. There are two outputs G and P of the next stage which are generated. The G of the next stage is generated by first taking the P of present stage and G of previous stage to go for an AND operation and giving out an output as Y taken as wire and which acts as an input to the OR gate which has an second input as G of the present stage giving out the final output which is G of the next stage and P is obtained by ANDing the P of previous stage and P of the present stage. The waveforms were obtained and verified using the test bench. And_xor: and_xor module has been implemented. It is the very first step in the kogge stone adder addition. The input A and B which we will receive will first be XORed together to give the output as G and ANDed together to give the output as P. these are the outputs which we have used 26 in the Black Cell module. The simulated waveforms were obtained and were verified according to the test bench.

Grey Cell: The gray cell module is the third and last module used for the kogge stone adder design. This module has 3 inputs and 1 output. One wire has been used in it as Y. the basic operation taking place here is the AND and OR operations. The 3 inputs are the G of present stage and previous stage and P of the present stage. This module has been simulated successfully.

Kogge Stone Adder: A kogge stone is parallel prefix adder and a type carry lookahead adder. This module will next be used in the Wallace tree multiplier's final stage computation. Instead of slow ripple carry adder which takes 9 stages to propagate and generate the carry the kogge stone adder is used which does all the computation in just 4 stages. Behavioral modeling style is used in verilog for the code. In which the inputs are taken as x , y and cin and outputs which are taken are sum and cout as any other adder the input x , y and output sum are of 8-bit each, cin and cout are the carry in and carry out signals and thus are of 1-bit each. Various wires are taken for the substages for the computation. The level 1 or the first stage conists of 1 gray cell and 7 black cells. Here the modules black cell and gray cell are used. They are called in the final module. The gray cell obtains its output from the initial cin value and black cell computes the G and P by the function discussed above. Now we move to the level 2, in this stage 5 black cell are used and 2 gray cells are called. Their output is further propagated to the next stages for the computation and the next is level 3 where 4 gray cells are used and only 1 black is used.

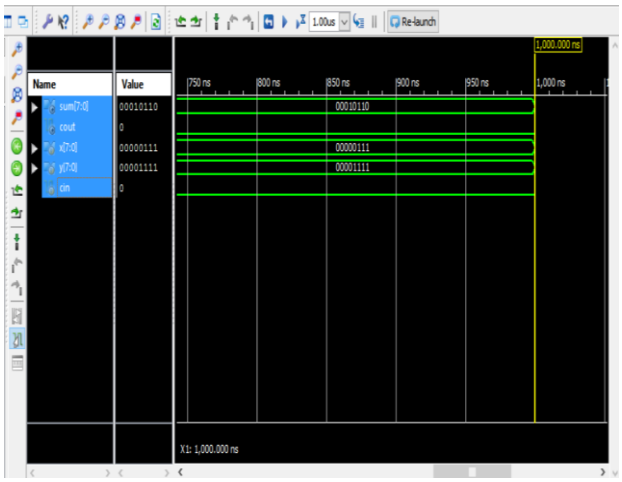

Fig 3: Simulation waveforms for kogge stone adder

```
Timing constraint: Default path analysis
  Total number of paths / destination ports: 180 / 9
--------------------------------------------------------------------------------
Delay:               12.385ns (Levels of Logic = 8)
  Source:            y<2> (PAD)
  Destination:       cout (PAD)

  Data Path: y<2> to cout
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             6   1.218   0.844  y_2_IBUF (y_2_IBUF)
     LUT2:I0->O            4   0.704   0.622  level_Z2/Mxor_p_Result1 (P_Z<2>)
     LUT4:I2->O            1   0.704   0.424  level_7D/G121_SW0 (N23)
     LUT4:I3->O            2   0.704   0.482  level_7D/G121 (level_3C/Y)
     LUT3:I2->O            1   0.704   0.424  level_7D/G13 (level_7D/G13)
     LUT4:I3->O            1   0.704   0.455  level_7D/G31_SW0 (N27)
     LUT3:I2->O            1   0.704   0.420  level_7D/G31 (cout_OBUF)
     OBUF:I->O                 3.272          cout_OBUF (cout)
    ----------------------------------------
    Total                     12.385ns (8.714ns logic, 3.671ns route)
                                       (70.4% logic, 29.6% route)

--------------------------------------------------------------------------------
```

Fig 4: Timing delay calculation of kogge stone adder

After this computation is done, we further move on to the level 4 or the stage 4 which is the last stage for the computation or calculation here 1 gray cell is used and and_xor module is called. 8 times the and_xor module is used because we have 8-bit input.

Moving further to obtain the sum we simply use the XOR inbuild in the verilog language and use the G, P obtained from previous stages to give the sum bit which is the output. 8 XOR gates are used here for the 8-bit output. The test bench is created next where various 8-bit values are given. The module was implemented successfully, and simulation result and waveforms obtained were verified according to the test bench.

III. RESULT AND CONCLUSION:

Table 1: Comparison of Wallace Tree Multiplier and Kogge Stone Adder results in terms of delay with previous results

Name Of The Digital Circuits	Previous Results (Delay in ns)	Improved Results (Delay in ns)
Wallace Tree Multiplier using carry look ahead adder	13.797	13.213
Kogge Stone Adder	15.160	12.385

This paper proposes improvement in the design of Wallace tree multiplier and kogge stone adder when considered with respect to the processing delay. By using half adder and full adder in partial product reduction stage alone with the carry look ahead adder at the final stage has proved to be advantageous in reducing the number of stage when compared with conventional Wallace tree multiplier. Also, the delay of Kogge Stone Adder has been minimized by optimizing the code.

REFERENCES:

1. C. S. WALLACE “A Suggestion for a Fast Multiplier” IEEE TRANSACTIONS ON ELECTRONIC COMPUTERS
2. Vinoth, C., Bhaaskaran, V. S. K., Brindha, B., Sakthikumaran, S., Kavinilavu, V., Bhaskar, B., Kanagasabapathy, M., and Sharath, B., “A Novel Low Power Wallace Tree Multiplier for RISC Processor”, IEEE 3rd International Conference on Electronics Computer Technology, Kanyakumari, pp. 330 – 334, 2011.
3. Khan, S., Kakde, S., and Suryawanshi, Y., “VLSI Implementation of Wallace Multiplier Using Energy Efficient CMOS Full Adder”, IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, pp. 1-4, 2013.
4. .S.Rajaram , Mrs.K.Vanithamani , “Improvement of Wallace multipliers using a fast adder” 978-1-61284-653-8/11/$26.00 ©2011 IEEE Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011)
5. Swartzlander, E. E., and Waters, R. S., “A Reduced Complexity Wallace Multiplier Reduction”, IEEE Transactions on Computers, vol. 59, pp.1134-1137, 2010.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)