Design of Area and Power Efficient Multiplier Unit using Wallace Tree Algorithm

R.Krishnaveni, P.Sakthy Priya, B.Sivaranjani, M.Sathish Kumar M.E., I.Vivek Anand M.E.

Abstract: ***In any processing system, the core data path element may be a multiplier that particularly associated with DSP applications, which concludes overall processing unit performance. In such system, the multiplier unit improves the performance will boost up the potential. The Cadence EDA performs for high-speed multiplier evaluation consisting 64 x 64 bit in ASIC Digital flow (RTL-GDSII). Using carry select adder and carry save adder, the proposed multiplier intend with Wallace structure for enhancing the speed criteria. Wallace rebate absorbs more time for designing because it is more complex and aberrant in format for better width. Real Time signal processor requires high productive capacity and fewer reaction time. A classic scheme could be today in IOT utilizations. The proposed method concentrates on designing low power and area efficient in digital flow using the Wallace tree algorithm. The 180 nm CMOS technology for pursuance and outcome are related with other existing methods in delay, area and dynamic power dissipation.***

Keywords: ***ASIC Digital flow, Cadence EDA, Carry select adder, Wallace tree structure.***

I. INTRODUCTION

Multiplication is the basic mathematical process which is widely used for calculation. Most of the arithmetic operations require the multiplication for implementation. Real Time signal processor requires high productive capacity and fewer reaction time. A classic scheme could be today in IOT utilizations. The present progress within the intent processors speed is at high. Then the necessary of high speed multipliers increased. Therefore, a designer concentrates maximum on circuit design's speed. Commonly, the interpretation of any DSP processor is torment from the sort of multiplier used.

Multiplier advancement is finite by reduction and addition of partial products. There are contrasting models and planning for designing a Multiplier. Wallace Tree Structure is executive than alter architect with respect to performance and features. Wallace Multiplier fascinates less power and its exchanging speed is quicker when correlate to other multiplier column. As a results of different aesthetics are proposed to style a far better Wallace multiplier architecture.

Revised Manuscript Received on May 06, 2020.

***** Correspondence Author

R.Krishnaveni, Electronics and Communication Department, National Engineering College, Kovilpatti, India, Email:krishnavenir187@gmail.com

P.Sakthy Priya, Electronics and Communication Department, National Engineering College,Kovilpatti,India,Email:priyabutterfly1998@gmail.com

B.Sivaranjani, Electronics and Communication Department, National Engineering College, Kovilpatti, India,Email:ranjanishiva1999@gmail.com

M.Sathish Kumar M.E., Assistant Professor**,** Electronics and Communication Department, National Engineering College, Kovilpatti, India, Email:Sathish27.ece@gmail.com

I.Vivek Anand M.E., Assistant Professor**,** Electronics and Communication Department, National Engineering College, Kovilpatti, India, Email: ilangovivek@gmail.com

In present days, most of the processing units are of 64 bits. Even though we can have 128 bits or more processing unit, which has maximum number of times to spend more silicon area and other key aspects successively increasing the value of hardware. More instruction sets grant to act arithmetic on discretionary width and may not be for a limited part of volume.

For that reason, we introduce a odd manner for advancement of multiplier unit uses Wallace for partial products contraction for 64 bits.

II. EXISTING ARCHITECTURE

Different multiplier architectures are suggested prior with the with the motive of clarifying multiplier performance. Few speedy multipliers includes Booth Multiplier, Modified Wallace reduction techniques with different types of adders viz., carry look ahead adders[6], Kogge Stone Adder, Brent Kung Adder and compressors such as 3:2, 4:2, 5:2 methods[5], Vedic mathematics using Redundant Binary Coding method[1] and sutras[4].

A. Wallace Tree Reduction

Tree multipliers are one type of approach in computing fast reduction of partial products. Wallace reduction is one of the efficient approach which improves speed of multiplication with higher width operands. Conventional Wallace tree is an prominent method to scale back partial product s. C. S Wallace proposed the method in 1964. This method will be more prominent for multipliers more than 16 bits. Partial product arrays are reduced with column based addition using counters. There may be a different size in counters. The final product is computed by unit of final adder. This process is established for 8 bit by 8 bit Wallace tree depicted in dot representation of reducing partial products.

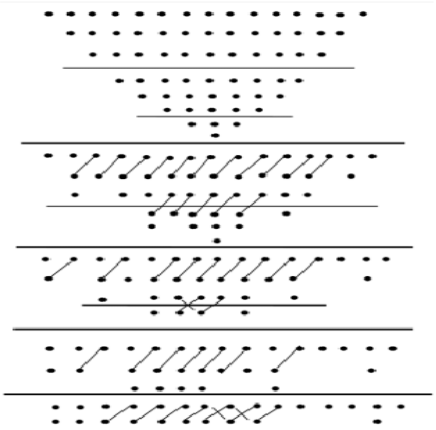

Fig.1 Conventional 8 bit Wallace Tree

III. PROPOSED ARCHITECHTURE

4x4, 8x8 multipliers of existing method has cost-effective and erratic design with final state RCA adder with average 3:2 compressors [5].When we go for higher adders the time is increases due to complexity and irregularity. Partial product is divided by proposed method into 4 equal arrays. With the proposed method the multiplication of 8bit multiplicand 8bit multiplier is performed.

With developed Wallace reduction by using counters each array of partial products is figure out. The result is 4 intermediate product terms, which is consecutively added with fast adders by considering correct bit position. For fast addition to meet out criteria of gain speed carry Select adder is used[3] . In parallel all four arrays are computed in single stage by using this proposed method, four intermediate product terms after that will get final product in 3 stage. The above approach is integrated for higher bits using different Wallace bits. Proposed architecture precisely point outs reuse of the planning to create units of high which reduces design complexity and better regularity.

A. Carry Select adder

Comprises of Ripple carry adders and a multi1plexer to pick proper sum and perform . For adding the bits with 0 as carry input and 1 as carry input two additions are used. When carry arrives from previous block then final sum and perform is chosen using multiplexer. This technique provides speed additionally and uses more resources. In carry select adder the block size can be fluctuate or unique..Example of 16bit carry select adder is illustrated in fig -2 with uniform block size of 4 bit.

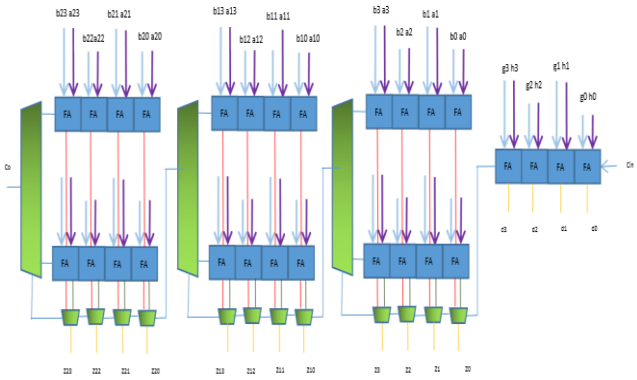

Fig.2 16 bit Carry Select Adder

IV. ASIC REALIZATION

The work is achieved for,

- Conventional Wallace Tree multiplier (8-bit).
- Proposed method with CPA (8, 16, 32 and 64-bit).
- Proposed method with CSA (8, 16, 32 and 64-bit).

A. Methodology

In this methodology, we present ASIC recognition with carry select adder for 64-bit Wallace Tree multiplier. Conventional Wallace tree (8-bit) and proposed method with carry propagate adder for 8, 16, 32, 64 bits results are used[2]. For the comparison. Results of conventional Wallace method and proposed method with carry propagate adder for 8,16,32,64 bits are used.

1) RTL design

By using cadence 45 nm technology with Verilog HDL the design is implemented .The functionality of the design was verified by using random function for engender random test vectors with VHDL test bench code. Fig-3 represents the complete design flow followed, alongside EDA Tool was used.

Cadence NC Simulator used for the simulation of designed multipliers. To achieve the maximum code average Cadence ICCR tool is selected. Simulation results are shown depicting 64 bits random inputs bring out from test bench and in hexadecimal and the RTL schematic after discussion of the proposed design

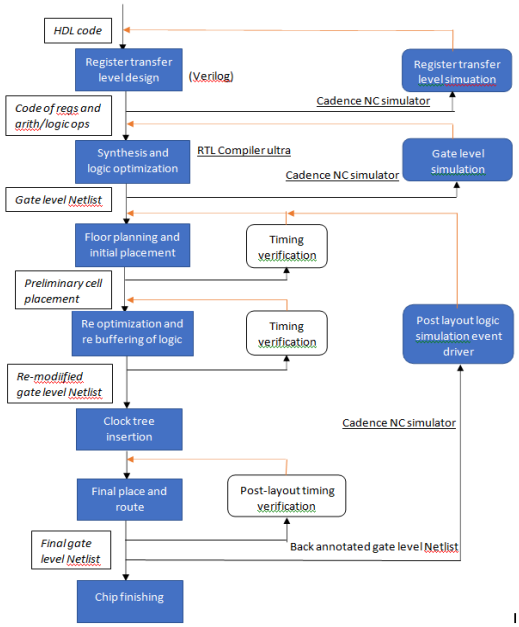

Fig.3 Digital design flow for ASIC realization

2) Synthesis

By using cadence RTL Compiler, Gate level synthesis was performed. With gate level simulation, formal and logical equivalence of the Netlist was verified and generated . Standard Delay Format (SDF), library files, test benches and Netlist files were used by GLS. For the proposed designs, to consume area instances were used. Fig.10 presents the power report and Fig-8 presents the timing report that shows the detailed timing report from RTL compiler with feeble information Using slow library, synthesis was administered by medium effort. In RC encounter gtd report conveys the synthesis for timimg debug .

3) RTL to GDS-II:

For physical style, Cadence Encounter tool was employed. We tend to proceed in physical style, once the gate level verification in symphonized Netlist. Atomic number 46 uses many files such as QRC school files, sdc file, Netlist. MMMC tool was employed for temporal arrangement analysis that is designed with Delay corner analysis beside setup and hold analysis.

(i) Floor planning:

Floor planning method is one of the method to create segments in core space and to make difference for every cells. For creating design, die space is calculated. Power rails were arranged victimisation metal ten and metal eleven

(ii)Placement and Routing:

Placement and routing is one of the method to distribute by placing it in exact places to plain cells above the chip without overlying. Temporal order can also be checked during this process. Improvement is allotted when routing just in temporal order.

iii)Clock Tree synthesis:

CTS is the synthesis to insert buffers to create certain clock inputs within style which receives clock with delay to avoid skew. Temporal order were in serious trouble before CTS. WNS is the planning process to optimize the sloppy issue in the mean time for the style and to verify for any geometrical or property violations and also the effect occurs in antennas. For additional post layout simulation, RC eradication is organized and so the Netlist is guarded

(iv)Post Layout Simulation:

It is the method to check the the practicality of the Netlist when CTS method with layouts inserts further components to style.

(v)GDS II:

To generate IC masks, GDS II is the format

V. RESULT

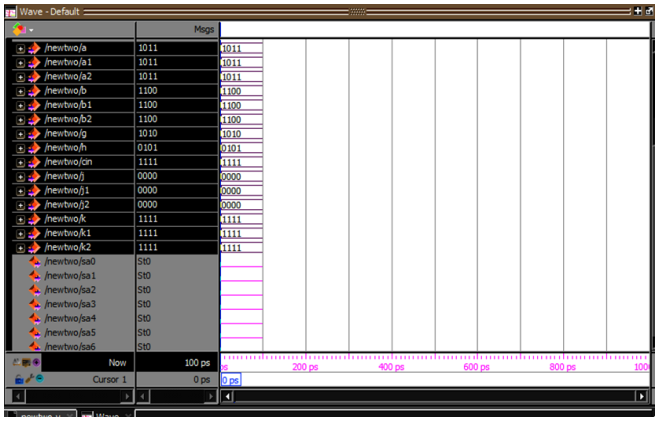

Fig.4 Waveform for Multiplier unit

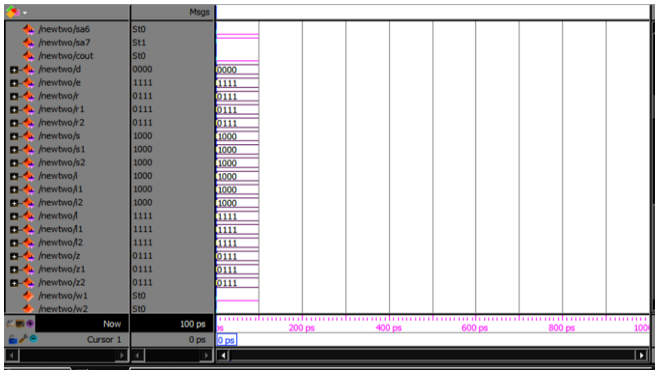

Fig.5 Waveform for Multiplier unit

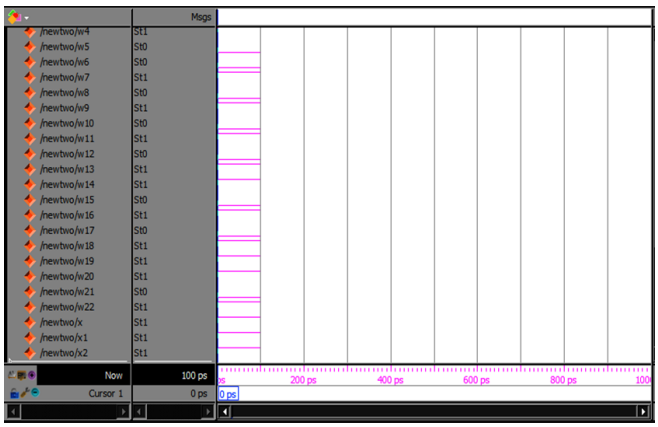

Fig.6 Waveform for Multiplier unit

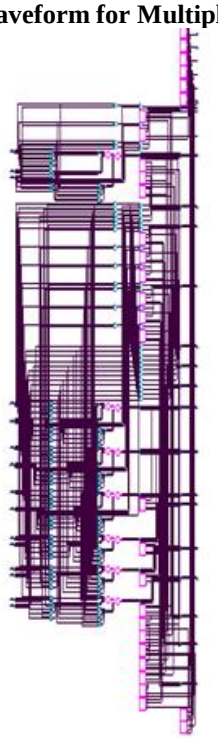

Fig .7 RTL View of Multiplier unit

```
Generated by:           Encounter(R) RTL Compiler RC10.1.304 - v10.10-s339_1
Generated on:           Feb 07 2020  11:58:08 am
Module:                 newtwo
Technology library:     slow_normal 1.0
Operating conditions:   slow (balanced_tree)
Wireload mode:          enclosed
Area mode:              timing library
============================================================

  Pin              Type        Fanout Load Slew Delay Arrival
                                      (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------
b2[1]       (u)  in port             4 20.0    0    +0       0 F
g249/in_1                                       +0       0
g249/z      (u)  unmapped_and2       4 20.0    0   +62      62 F
add_25_36/A[1]
  g6/in_0                                       +0      62
  g6/z      (u)  unmapped_nor2       1  5.0    0   +38     100 R
  g12/in_0                                      +0     100
  g12/z     (u)  unmapped_not        2 10.0    0   +37     136 F
  g24/in_0                                      +0     136
  g24/z     (u)  unmapped_nand2      1  5.0    0   +38     174 R
  g25/in_1                                      +0     174
  g25/z     (u)  unmapped_xnor2      2 10.0    0   +94     268 R
add_25_36/Z[1]
add_25_54/A[1]
  g6/in_0                                       +0     268
  g6/z      (u)  unmapped_nor2       1  5.0    0   +38     306 F
  g12/in_0                                      +0     306
  g12/z     (u)  unmapped_not        2 10.0    0   +37     343 R
  g13/in_1                                      +0     343
  g13/z     (u)  unmapped_nand2      1  5.0    0   +38     380 F
```

Fig.8 Detailed timing report from RTL Compiler

```
tance is calculated
state_retention        - prints state retention report
summary                - prints an area, timing, and design rules report
test_power             - estimates the average scan power in shift and capture m
odes during test
timing                 - prints a timing report
units                  - reports units
yield                  - prints a yield report
rc:/> report area
============================================================
  Generated by:           Encounter(R) RTL Compiler RC10.1.304 - v10.10-s339_1
  Generated on:           Feb 07 2020  11:56:48 am
  Module:                 newtwo
  Technology library:     slow_normal 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Instance   Cells  Cell Area  Net Area   Wireload
-----------------------------------------------------------
newtwo         424       2379         0     <none> (D)
  add_20_54     22        120         0     <none> (D)
  add_20_36     22        120         0     <none> (D)
  add_19_54     22        120         0     <none> (D)
  add_19_36     22        120         0     <none> (D)
  add_25_54     22        120         0     <none> (D)
  add_25_36     22        120         0     <none> (D)
  add_24_54     22        120         0     <none> (D)
  add_24_36     22        120         0     <none> (D)
  add_15_49     22        120         0     <none> (D)
  add_15_33     22        120         0     <none> (D)
  add_14_49     22        120         0     <none> (D)
  add_14_33     22        120         0     <none> (D)
  add_10_51     22        120         0     <none> (D)
  add_10_33     22        120         0     <none> (D)

 (D) = wireload is default in technology library
rc:/>
```

Fig.9 Area Report

```
  Generated by:           Encounter(R) RTL Compiler RC10.1.304 - v10.10-s339_1
  Generated on:           Feb 07 2020  11:57:09 am
  Module:                 newtwo
  Technology library:     slow_normal 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                    Leakage   Dynamic    Total
  Instance  Cells Power(nW) Power(nW) Power(nW)
-------------------------------------------------
newtwo        424  3512.013 42135.473 45647.486
  add_10_33    22   140.970  1292.568  1433.538
  add_10_51    22   140.970  1408.835  1549.805
  add_14_33    22   140.970  1442.430  1583.400
  add_14_49    22   140.970  1504.440  1645.410
  add_15_33    22   140.970  1589.614  1730.584
  add_15_49    22   140.970  1715.169  1856.139
  add_24_36    22   140.970  1366.836  1507.806
  add_24_54    22   140.970  1487.238  1628.207
  add_25_36    22   140.970  1317.130  1458.099
  add_25_54    22   140.970  1434.088  1575.058
  add_19_36    22   140.970  1370.814  1511.784
  add_19_54    22   140.970  1367.427  1508.397
  add_20_36    22   140.970  1341.315  1482.285
  add_20_54    22   140.970  1246.180  1387.150
  m1            0     0.000   601.172   601.172
  m2            0     0.000   622.266   622.266
  m3            0     0.000   622.266   622.266
  m4            0     0.000   611.719   611.719
```

Fig.10 Detailed Power Report

Table-I: Comparison of 64bit multiplier with proposed and existing method

Multiplier	No. of Bits	Delay	Area in Cells	Power	Frequency
Existing CPA	64	22.341 ns	6342	0.6559 mW	45.1 MHz
Proposed CSA	64	94 ps	2379	45647.486 nW	10.6 GHz

VI. CONCLUSION

Wallace reduction is a competent technique to implement high speed multiplier architecture. To implement successive designs, the designs were reused which relies on the proposed method that exposes the evenness. The traditional or existing method in the complex irregular structure was eliminated. Using carry select adder it was achieved that the performance was better with respect to speed and experimental results were shown for 64 bit Wallace multiplier with proposed method-3. The delay attained is 94ps with the manageable frequency of 10.6 GHz .It is also been observed that proposed architecture inclines more notable to realize high speed estimation for 64 bits and better multipliers. In any VLSI Design, the parameters such as Area, Power and Timing were always been a trade off. Multipliers are often administered with optimized adders.

REFERENCES

1. Keote, R. S., & Karule, P. T. (2016). VLSI design of 64bit× 64bit high performance multiplier with redundant binary encoding. In 2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall) (pp. 1-6). IEEE.
2. Chaithra, T. G., PradeepKumar, S. K., Nandini, K., Harshitha, S. V., & Ankitha, K. R. (2017, May). ASIC realization and performance evaluation of 64× 64 bit high speed multiplier in CMOS 45nm using Wallace Tree. In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 1115-1119). IEEE..
3. Ramesh, G., & Lakshmi, K. N. A NOVEL WALLACE TREE MULTIPLIER FOR USING FAST ADDERS.
4. Jais, A., & Palsodkar, P. (2016, April). Design and implementation of 64 bit multiplier using vedic algorithm. In 2016 International Conference on Communication and Signal Processing (ICCSP) (pp. 0775-0779). IEEE.
5. Vaishnav, M., & Tech, M. (2012). Design of multi-precision reconfigurable Wallace Tree Multiplier for high performance applications. International Journal of Information, 1(1).
6. Vashisht, P. G., & Dadhich, P. (2015). Design and Analysis of 64 bit Multiplier using Carry Look Ahead Logic for Low Latency and Optimized Power Delay Product. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 4, 7847-7853.

AUTHORS PROFILE

R.Krishnaveni is an undergraduate student in department of Electronics and communication engineering at National Engineering College, Tamil Nadu, India. Her research interest includes VLSI and networking

P.Sakthy Priya is an undergraduate student in department of Electronics and communication engineering at National Engineering College, Tamil Nadu, India. Her research interest includes VLSI and networking.

B.Sivaranjani is an undergraduate student in department of Electronics and communication engineering at National Engineering College, Tamil Nadu, India. Her research interest includes VLSI and networking

M.Sathish Kumar received the B.E degree in Electronics and Communication Engineering from Hindustan College of Engineering (2009) Chennai,India, M.E degree in Communication Systems from National Engineering College(2011) Kovilpatti,India,and currently pursuing his Ph.D in the area of Device Modeling

I.Vivek Anand received B.E degree in Electronics and Communication Engineering from Sethu Institute of Technology (2012) and M.E degree in VLSI from Mepco Schlenk Engineering College (2014). He is currently working in National Engineering College, Kovilpatti, India as Assistant Professor. His research interest includes Modeling and Simulation of Tunnel Field effect Transistors.