Abstract—A multiplier is one of the most important building block that is widely used in processor, embedded systems, application specific integrated circuits and most of the DSP applications. Low power is an emerging trend which intern can maximize the lifespan of battery operating time. In this project, it is proposed to balance and optimize the performance of Wallace multiplier which consumes less power. The two main sources of power consumption are static power dissipation and dynamic power dissipation. The multiplier has been designed and simulated using cadence tool. The three main thrust parameters of any VLSI design lies in speed, area and power. Low power is an emerging trend which intern can maximize the lifespan of battery operating time. The main objective of the project is to design and implement a low power multiplier used for various VLSI applications. The work includes designing of basic gates, half adder and full adder with operating voltage. The multiplier block is implemented using Microwind tool. The logic styles used in our proposed design of the multiplier are CMOS. The power analysis has been carried out and measured on CMOS logic.

Keywords— Multiplier algorithm, Wallace multiplier, CMOS logic, Half adder, Full adder.

I. INTRODUCTION

Now a days, Multiplier is one of the most important blocks in any processor. A binary multiplier is an electronic circuit used in digital electronics, to multiply two binary numbers. A variety of computer arithmetic techniques can be used to realize a digital multiplier. Most techniques involve computing a set of fractional products, and then summing the fractional products together. This process conducting the long multiplication on base-10 integers, but have been modified as a number of base two systems. In more transistors, per chip became available due to larger-scale integration. It became to put enough adders on a single Chip to sum all the fractional products at once than use again a single adder to handle each partial product one at a time. In digital signal processing, algorithm spends more time to multiply the processors. It spends a lot of chip area in order to make the multiplication as fast as possible. Hence a non conventional however very efficient Vedic mathematics is used for making a high performance multiplier. Vedic algorithms deals mainly with various Vedic mathematical formulae and their applications for carrying out large arithmetical operations easily. To evaluate the performance of the new multiplier, the multiplier is compared with the already existing digital multipliers on various parameters as power consumption and speed of operation. In a typical processor, central processing unit involves considerable amount of processing time in performing arithmetic operations, particularly multiplication operations. Multiplication is one of the basic arithmetic operations. It requires more hardware resources and processing time when compare to addition and subtraction. MULTIPLIERS can be designed by using any of the following methods. Here the main objective is to design low power consumption, consuming less area and a high-speed multiplier.

II. LOW POWER MULTIPLIER UNIT

A. Low Power Multiplier Unit

In any signal processing system, Multiplier plays an important role and also perform as a basic building block element. The performance of these types of processing systems depends on the performance of the inbuilt multiplier. So it is acts as a challenging task for any designer to design a high performance multiplier. There are different factors that drive high performance electronic system for to design in terms of low power dissipation and high speed. The basic block diagram of a multiplier is shown in Fig 1. A basic multiplier consists of three stages:

1. Generation of partial product
2. Addition of partial product
3. Final addition

![Fig-1: Block diagram of multiplier unit](image)

B. Multiplier Algorithm

Array Multiplier
Array multiplier is a traditional method for multiplication. Array multiplier is popular due to its structure. It is based on add and shift algorithm. In parallel multiplication, the number of partial products to be added is the main parameter that determines the performance of the multiplier. With one multiplier bit each partial product is generated by the multiplication of the multiplicand. The partial products are shifted according to their bit order and then it gets added with normal carry propagate adder.

For $n \times n$ array multiplier, number of adders and gates required are:

1. $n(n-2)$ full adders
2. $n$ half adders
3. $n^2$ AND gates

The advantage of array multiplier is that it has minimum complexity and regular structure. Disadvantages are large number of logic gates, so more chip area and it has high power consumption and it is limited to 16-bits.

Wallace-Tree Multiplier

Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers, devised by Australian Computer Scientist Chris Wallace in 1964. The Wallace tree has three steps:

- In one of the arguments each bit is multiplied by each bit of the other, yielding $n^2$ results. The wires carry different weights depending on position of the multiplied bits.
- The number of partial products reduced to two by layers of full and half adders.
- The wires in two numbers gets grouped, and added with a conventional adder.

The advantage of Wallace-tree multiplier is that it becomes more pronounced for more than 16-bits. And Disadvantage is that a logarithmic depth reduction tree-based CSA’s has an irregular structure, therefore its design and layout is difficult.

Booth Algorithm

Booth algorithm is the multiplication algorithm. It multiplies two binary numbers in two's complement notation of signed binary numbers. The algorithm was invented by Andrew Donald Booth in 1950 while doing research on Crystallography at Birkbeck College in Bloomsbury, London. Booth's algorithm is the interest of studying computer architecture. Booth multiplier is best for signed numbers. Booth used desk calculators that were faster at shifting than adding & created the algorithm to increase their speed. Hence to reduce the iterations Booth’s Algorithm is developed for multiplying signed as well as unsigned numbers. It initiates with the ability to both add and subtract, there are multiple ways to calculate a product. This multiplier can scan the three bits at a time hence the delay decreases. But the power consumption of this multiplier is more hence the efficiency of the system reduces.

Example: $3 \times (-4)$.

$m = 3 = 0011$, $r = (-4) = 1100$.

Multiplication can be implemented by repeatedly adding one of two predetermined values A and S to a product P, then rightward arithmetic shift on P.

Karatsuba Algorithm

The Karatsuba algorithm is a fast multiplication algorithm. It was discovered by Anatoly Karatsuba in 1960, published in 1962. Karatsuba algorithm uses a divide and conquers approach. Where it breaks down the inputs into Most Significant half and Least Significant half.

Recursive application of Karatsuba Algorithm

If $n$ are four or more, the three multiplications in Karatsuba's basic step involve operands with fewer than $n$ digits. Therefore, products can be computed by recursive calls of the Karatsuba algorithm. The recursion can be applied until the numbers are small that they can be computed directly.

In a computer with a full 32-bit by 32-bit multiplier, for example, one could choose $B = 231 = 2,147,483,648$, and store each digit as a separate 32-bit binary word. Then the sums $x1+ x0$ and $y1 + y0$ will not need an extra binary word for storing the carry-over digit, and the Karatsuba recursion can be applied until the numbers to multiply only one digit long.

Karatsuba algorithm uses divide and conquer approach where it break down the inputs into Most significant half and Least significant half. Karatsuba algorithm is suited for operands of higher bit length. For multiplication, break down the input into two such as $XH$ and $XL$.

Vedic Multiplier

Vedic Mathematics is a book written by the Indian monk Swami Bharati Krishna Tirtha, published in 1965. It contains a list of mental calculation techniques claimed to be based on the Vedas. The calculation system mentioned in the book is also known by the same name or as "Vedic Maths". It is characterised as "Vedic" mathematics and has been criticized.
by academics, who have also opposed its inclusion in the
Indian school curriculum. Ancient mathematics has 16
different sutras, which are taken from Atharva Ved. For
multiplication, there are two sutras. Urdhva-Tiryagbhyam is
one of the sutra from 16-Vedic sutras which performs the
product of two decimal numbers. Urdhva-Tiryagbhyam is the
general formula applicable to all cases of multiplication of a
large number by another large number. “Urdhva” means
vertical and “Tiryagbhyam” means crosswise therefore it is
also called as vertical and Crosswise Algorithm.

Its advantage is that it has Minimum Delay. As number of
bits increases, multiplication process becomes tedious which
is a major disadvantage in Vedic multiplier.

By comparing all the algorithms. We prefer Wallace tree
multiplier because it provides the only efficient multiplier
which has substantial hardware savings, higher speeds, less
propagation delay, has reduced schematic layout and occupies less area. Hence Wallace is very often preferable for
multipliers.

III. PROPOSED WALLACE TREE MULTIPLIER

A. Wallace Tree Multiplier

A Multiplier which is based on Wallace-tree structure is
called Wallace multiplier. It is more faster than other
multiplier architecture. Wallace multiplier operation is
carried out in three different steps. In this architecture, after
generating the partial product, accumulation of partial
product and final addition are done in different stages. When
the final stage contains only two rows, then final addition is
done. The number of rows of partial product in a particular
stage can be expressed as,

\[
R_{i+1} = 2 \left( \frac{R_i}{3} \right) + R_i \mod 3
\]

where, \( R_i \) gives the groups or stages. \( N \) = number of
bits. Let us consider an example that \( N \) bits multiplication,
\( N \) \( \\text{AND} \) gates are required to generate the partial product
terms and the number of reduction stages is given by

\[
S = \log_2 N
\]

Basic Building Blocks of Multiplier

- Formation of partial product using \( \text{AND} \) gate logic.
- Reducing the ‘n’ number of partial products to a
two-row partial products by compressing the
column’s with \([3,2]\) \& \([2,2]\) adders.
- Merging two-rowed partial products with carry
propagation Adders.
- 2bit result.

Function and Algorithm of Modules

2’S Complement Generator

Function: The 2’s complement generator takes the
multiplicand MD and MR as its input and produces MD and
–MR as its output in case of negative numbers.

Algorithm: 2’s complement is generated by inverting all
bits of the multiplicand and then adding 1 using a ripple carry
adder.

Partial Product Generator

Function: The partial product generator generates the
partial products to be added with a Wallace tree.

Algorithm: The partial product generator uses the table
for each multiplier bit. Depending on the value of MD or –
MD and MR or–MR, it is assigned to partial product. 4 bit is
then extended to 7th bit for appropriate sign extension.

Carry Look-Ahead Adder

Function: Carry Look-Ahead adder (CLA) add two
numbers with very lower latency.

Algorithm: By extending c with the corresponding
inputs, the carry and sum are independent of the previous
bits.

Wallace Multiplication

Function: The Wallace tree module adds with the 4
partial products and generates two intermediate operands for
final addition.

IV. SOFTWARE USED

MICROWIND is truly integrated EDA software of IC
designs from a concept to complete by enabling chip
designers to design beyond their imagination. It integrates
traditionally separated front-end and back-end chip design
into one flow, accelerating design cycle and reduces design
complexities. It tightly integrates mixed-signal implementation with digital implementation, circuit
simulation, transistor-level extraction and verification to
provide an innovative education and initiative to help
individuals, to develop the skills needed for design positions
in every domain of IC industry.

MICROWIND supports entire front-end to back-end
design flow. We have DSCH (digital schematic editor) which
posses in-built pattern based simulator for front-end
designing. User can also build analog circuits and convert them into SPICE files and use third party simulators like WinSpice or pSPICE.

DSCH can convert digital circuits into Verilog file which can be further synthesized for FPGA/CPLD devices of any vendor. In MICROWIND, the same Verilog file can be compiled for layout conversion.

MICROWIND supports the back-end design of circuits. User can design digital circuits and compile using Verilog file. MICROWIND automatically generates an error free CMOS layout. Although this place-route is not optimized enough as we do not indulge in complex place & route algorithms.

User can also create CMOS layout of their own use, compiling one line Verilog syntax or custom build the layouts by manual drawing. This layouts can be verified using inbuilt mix-signal simulator and analyzed for DRC, crosstalks, delays, 2D cross section, 3D view, etc.

V. RESULTS

![Fig-5: Schematic diagram of AND gate](image)

![Fig-6: Schematic diagram of OR gate](image)

![Fig-7: Symbol of Full adder](image)

![Fig-8: Schematic diagram of EXOR gate](image)
VI. CONCLUSION

Thus schematic is designed for four bit multiplier for which we provide two four bit data input lines and obtain the product of 8bit. The designed multiplier is tested for various data and functionally verified. The multiplier is implemented using CMOS logic style. The Wallace tree multiplier is designed and implemented using Microwind tool. The schematic is designed using DSCH. Microwind tool is used for the implementation of the multiplier design. The schematic is designed for 4-bit multiplier. The complete schematic of 4-bit multiplier is functionally verified and implemented using Microwind tool. The synthesis result confirms that the proposed Wallace tree multiplier is suitable for low power and small area applications. A low power and efficient multiplier is designed and implemented for various VLSI applications.

REFERENCES