Design and analysis of High-Speed Low-Power Vedic Multiplier with 3-1-1-2 compressor Using Reversible Logic gates

Ms.Dharani S1, Mr.Abin Satheesan3, Asuvanti M A3, Ranjith Kumar4, Vidhya S5

2 Assistant Professor, Dept. of ECE, Bannari Amman Institute of Technology, Tamil Nadu, India
1 3 4 5 PG Scholar, Dept. of ECE, Bannari Amman Institute of Technology, Tamil Nadu, India
E-mail: Dharnisubramaniam273@gmail.com

Abstract. The FFT Function in digital signal processing is one of the most important function in several applications such as Image Processing, Wireless Communications and Multimedia. FFT Processors consisting of butterfly structure operations involving necessary operations such as Addition, Subtraction, and Multiplication of complex values. The FFT Butterfly Structure work is designed with a “Vedic Multipliers” for applications at high speed. In this Vedic Multiplier, an algorithm called “Urdhva Triyabhyam” was used to improve its efficiency by optimizing the number of logic gates, constant inputs and garbage outputs. The Data Computation time is reduced by an 3-1-1-2 compressor using reversible logic gates. Hence reducing the surplus power consumption of 11.24% and summation of the partial products is done with less delay factor of about 5.28%. The area, power, delay, area delay product and power delay product are calculated using cadence virtuoso and is implemented in Spartan-6 device family using Xilinx ISE.

Keywords: Area; power; delay; Vedic multiplier; reversible logic gates.

1. Introduction

In forth coming generation, Wireless technology and technology (IOT) will play a crucial role in digital era. New access structures for next generation digital systems are the FDMA multi-access frequency division, OFDMA, “single carrier FDMA”, and “interleaved FDMA”. These programs are based on effective, Fast-Fourier transforming algorithms consisting of fewer arithmetic operations. The present-day circuits dissipating power of order of nano watts which is comparatively 3-4 magnitude larger than the power lost due to entropy change. The high-speed data processing system for several multimedia applications with lower power consumption and area is the most critical limit. The adder and multiplier are basic arithmetic unit most commonly used essential operation. The digital world of the next generation expects high quality digital information with high accuracy. For example, the system requires high-quality images. To satisfy the quality of image criteria pixel value is increased. Meanwhile increasing the computation time in the digital system.
2. Literature Survey

The configurable power multiplier with booth algorithm, dynamic range, adjuster and partial product generator corrections. This powerful design detects redundant calculations and reduces power consumption, more areas and the partly producing product.

The architecture of the multiplier with a fixed width consists of modified booth encoders, an error reward circuit and partial generation of products, which meet the needs for area and power consumption with a high degree of precision for lossless application with limited design complexity. Approximate power efficiency which includes a bit pre computation reselect and 2-1 mux. The presets are used to choose anyone from MSB or LSB. The device that achieves 99% precision in the application of audio and recognition.

The construction of approximately 15-4 compressors using 5-3 circuit is designed to reduce power and delay and increases the throughput. The construction of approximately 15-4 compressors using 5-3 circuit is designed to reduce power and delay. This concept is most commonly used in the image processing framework and the result shows improved performance.

In order to reduce the combinational time in a compressor 3-2, 4-2, 5-2, the XNOR-XOR logic is implemented. 4-2 compressors remain as high-speed multiplexers to decrease the time-consuming delay in the summation part of the product and are constructed using XNOR-XOR gates and 2:1 mux. 3-1-1-2 compressor consisting of a reduced area and less delay by summing up the partial products.

3. Existing System

3.1 Vedic Mathematics

Vedic maths is an assortment of procedures to illuminate scientific mathematics in simple and quicker manner which can be utilized for issues involved in arithmetic, algebra calculus, polynomial math, geometric analytics, conics and so on. Vedic mathematics is a method that Indian mathematicians have developed to solve problems with standard mathematical measures. Often complex and time consuming, but using general Vedic mathematical principles and particular techniques, numerical calculations can be carried out very quickly. Vedic science is an arithmetic framework which was found by Indian mathematician utilizing standard numerical advances, considering various issues.

Advantages
- Vedic arithmetic can definitely solve scientific, numerical figuring in quicker manner
- More than 1700 per cent faster than ordinary math
- The procedure Speeds up and performance
- It improves scholastic execution and instant moment results

3.2 Nikhilam Navatashcaramam dashatah

For the concept of "computing quickly." If you say the number of bit-wise multiplications (and additions) is reduced, the answer is that they don’t. Both methods take multiplications and additions around $n^2$ bit-wise, where $n^2$ is the number of bits in the multiplicand and the multiplier each. Unless you mean the need for memory then the sutra would be significantly stronger. You first write down $n^2$ intermediate rows in the general method, and move them column by column, from right to left. This method is one of the 16 sutras from Vedic mathematics. This sutra can be used to simplify multiplication of large digit numbers into smaller numbers with only a few subtractions, adds, and shifts, the steps are followed below:

- We pick the power of 10 as our foundation depending on the numbers closer to the power of 10
- Apply the nikhilam theory, eliminate the digit from the base-all the digits 9 and 10 last. Such numbers are referred to as excess if positive and if negative, as deficiencies
- Apply cross addition to multipliers deficiencies or excess results
• The second stage must also have the same number of digits as the number of the chosen base's zeroes if zeroes are less than pre-appended values. If more, pass the initial numbers to the 1st stage.

Advantages
• Converting large numbers to corresponding small digit multiplication.
• Efficient if multiplier and multiplicand are similar to the same base power.

3.3 Urdhva Triyabhyam

Urdhva Triyabhyam is the popular and well-established equation suitable for all occurrences of multiplication and moreover in the division of an enormous value by another enormous value. It signifies "vertically and crosswise".

Steps:
For non-equidigit numbers, order the numbers vertically, prefix the same number of zeroes in front.
• Consider arrangements of vertical segments, preferably from left to right, increasing in order and raising in order afterwards.
• Recognize the sum of the edge-equidistant digit cross-results to show each sequence.
• Reformat the number in base 10, whenever possible. Sets are acquired by using each vertical section in turn from left to right (ascending) and then again with the exception of each vertical section in turn from right to left (descending).

For calculating n-digit set, the cross product is:
• For even number of digits: the entire number of digit outcomes, equidistant to the end.
• For the odd number of digits: the total number of digit tests, equidistant from the finishes applied with the primary digits multiplied by each other.

For a single-digit set,

For multiplying two-digit number, we can simply multiply two values, such as m1.m2, which brings cross-product of m1 x m2.

Advantages
• Easily realized in hardware
• Time taken to perform multiplication operation and design complexity of circuit is very less.
• Main advantage of using this type of multipliers reduces the number of the bits, area and speed gradually compared to the other fast multipliers.

3.4 Implementation of Vedic multiplier using urdhva Triyabhyam

Vedic multipliers are one of the fastest and efficient methods in the arithmetic function processing that Indian mathematicians developed and used. In this Vedic multiplication is logically expressed with “Boolean equation”. Bitwise summation equation is made a comparative list for 4x4 and 8x8 multiplier. In the summation equation ‘+’ symbol which represents the binary addition operation and it is executed by implementing and making use of half and full adders. The number of unwanted switching that can be reduced by using a detection unit. The detection unit for 4x4 and 8x8 input data are calculated by “2*(N/2-1) 2input logic OR gate”, “four 2 input AND gate”, and “2 NOT gate”.

Table 1. Detection unit logical gate count for N-BIT multipliers

| Multiplier N x N | No. of input AND gates | No. of input OR gates | No. of NOT gates | Total Required gates |
|------------------|------------------------|-----------------------|-----------------|---------------------|
| 4X4              | 04                     | 02                    | 02              | 08                  |
| 8X8              | 04                     | 06                    | 02              | 12                  |
3.5 Multiplier design with 3-1-1-2 compressor

The first stage compute is based on the bitwise logical AND operation of the 4x4 multiplier based on Vedic math’s. 16 logical AND gates from LSB to MSB are arranged to generate the single output in the first step. For the multiplication result S0 to S7 with carry propagation, the half adder’s and the full adder’s computation are made used. The first stage 16 “AND” gates are arranged based on input data type.

If the input date be X=00xx and Y=00xx data, only four logical AND gate operation is necessary during the 1st stage. Therefore, LSB positions these four logical AND operations from LSB. Also, when the X=00xx input data and Y = xxxx inputs 8 logical AND operation is required to replace the 16 logical AND operation for the following bit at the 1st stage of computation. Consequently, in 2nd phase and propagation phases half adder’s and full adder’s have been used. The 4x4 multiplier, based on 3-1-1-2 compressor, has a partial product propagated by the logical AND process. The 4x4 multiplier generates sixteen logical AND operations when the data is full-range, and it is ordered from 0 to 15.

Table 2. Logical equation for 4x4 multiplier with Detection unit

| Type of Multiplier | Input data type | With Detection unit |
|--------------------|-----------------|---------------------|
| 4x4                | X = 00xx ; Y=00xx | Z0 = x0y0; Z1 = x0y1 + x1y0; Z2 = x1y1 + c1; |
|                    |                |                     |
|                    | X = 00xx ; Y = xxxx | Z0 = x0y0; Z1 = x0y1 + x1y0; Z2 = x0y2 + x1y1 + c1; Z3 = x0y3 + x1y2 + c2; |

Figure 1. 4x4 Multiplier design with 3-1-1-2 compressor

4. Proposed Design

According to Landuer theory, Energy is dissipated if we have a logical operation on the unit. The cause for this operation is due to thermodynamic laws, in which entropy of system tends to increase whenever there is an operation that doesn’t have identical and different solution. This forms the basis of reversible logic concept where there will be correspondence between only input and output of the system. The one of the important weighted function in ALU is multiplication. The multiplier is the circuit which is tremendously used factor in many DSP applications. This multiplier block this involves two major process namely partial product generation and addition of partial products. Vedic Multiplier that offers two popularly used algorithm named “Urdhva Triyabhyam” and “Nikhilam” algorithm for multiplication. Here we use “Urdhva Triyabhyam” which is popularly used technique for fast multiplication by multiplying “vertically and crosswise”.


The algorithm is derived for 4x4 multiplication of binary numbers is given below:

\[ A_{0}.B_{0} = P_{0}.C_{0} \quad q_{0} \quad \[1\] \\
\[ (A_{1}.B_{0}) + (A_{0}.B_{1}) = P_{1}.C_{1} \quad q_{1} \quad \[2\] \\
\[ (A_{2}.B_{0}) + (A_{0}.B_{2}) + (A_{1}.B_{1}) = P_{2}.C_{2} \quad q_{2} \quad \[3\] \\
\[ (A_{3}.B_{0}) + (A_{0}.B_{3}) + (A_{2}.B_{1}) + (A_{1}.B_{2}) = P_{3}.C_{3} \quad q_{3} \quad \[4\] \\
\[ (A_{3}.B_{1}) + (A_{1}.B_{3}) + (A_{2}.B_{2}) = P_{4}.C_{4} \quad q_{5} \quad \[5\] \\
\[ (A_{3}.B_{2}) + (A_{2}.B_{3}) = P_{5}.C_{5} \quad q_{5} \quad \[6\] \\
\[ A_{3}.B_{3} = P_{6}.C_{6} \quad q_{6} \quad \[7\] \\

This may generate combinational delay in the circuit, which may decrease the speed of the system. In order to avoid delay in the circuit we introduce a block called compressor 3-1-1-2 using reversible logic. The reversible logic circuit is said to be reversible which is most popularly used to predict the input vector by output value this is because of previous value of information bit is not lost. The reversible logic is a new and promising field which reports the problem of power dissipation. Also proven to consume zero power theoretically.

The simple multiplication operation may consume many cycles to complete the operation. The main goal of embedded system is low power dissipation, high speed, less Area. There are various reversible gates available but we manipulate the fredkin gate which holds of quantum cost value of 5 and garbage value of only 3.

**Figure 2.** Fredkin gate (Reversible logic gate)
Computational circuit suitable for reversible computing consists of equal no. of input and output that permits the first bit to remain unchanged and reciprocate the last two bits only if the first bit is 1 and quantum cost of gate is 5.

The proposed circuit is mainly designed to compute the lower part of summation with three inputs and one carry input compressor are required to compute summation result with smaller combinational delay. It is a 3*3 "Fredkin Gate". If P=A, Q=C if A=1 else B, R=B if A=1 else C. Hence this gate is known as Controlled SWAP Gate (CSWAP). Quantum cost is 5 and its delay is 5. This gate is popularly known as Universal Reversible gate i.e., Any logical or Arithmetic operation can be performed entirely using Fredkin Gate. We can observe the drastic change, when every change of state two bits goes in and one bit comes out. Therefore, the above resistive losses of material, the gate must dissipate:

$$T_E = K \log 2T$$ Joules per transition. \[8\]

Where,
K - Boltzmann constant
T - Temperature

![Figure 3. Mux design using reversible logic fredkin gate](image)

In the above figure,

$$A = \overline{S_0}I_0 + S_0I_1$$ \[9\]

$$B = \overline{S_2}I_2 + S_2I_3$$ \[10\]

$$y = \overline{S_1}\overline{S_0}I_0 + \overline{S_0}S_0I_1 + S_1\overline{S_0}I_2 + \overline{S_1}S_0I_3$$ \[11\]

![Figure 4. Design of compressor 3-1-1-2 using Reversible logic Gate](image)

The above designed fredkin gate among other reversible logic gates to design mux which is efficiency made use to design 3-1-1-2 compressor using reversible logic gate is given in below

$$\text{Sum}=((n_1 \oplus n_2)\overline{n_3}n_4) + ((n_1 \oplus n_2)\overline{n_3}n_4) + ((n_1 \oplus n_2)n_3n_4) + ((n_1 \oplus n_2)n_3n_4)$$ \[12\]

$$\text{Carry} = (n_1 + n_2)n_3\overline{n_4} + (n_1 + n_2)n_3\overline{n_4} + (n_1 + n_2)n_3\overline{n_4} + (n_1 + n_2)n_3\overline{n_4}$$ \[13\]

$$C_{out} = n_1 \oplus n_2 \oplus n_3 \oplus n_4$$ \[14\]

The above equation has been proposed using k map and Boolean expression is derived.
The proposed 3-1-1-2 compressor achieved reduction in delay up to 11.98% compared to the previous existing compressor and also achieved reduced power dissipation, during the summation of partial products. 4x4 Bit multiplier is simulated using proposed “3-1-1-2 compressor” using reversible logic gates.

![Figure 5. 4x4 multiplier using compressor 3-1-1-2 with reversible logic gates](image)

**Output Waveform**

![Figure 6. Output waveform of 4x4 multiplier using compressor 3-1-1-2 with reversible logic gates](image)

### 4. Results and Discussion

The below chart that makes a perspicuous view about the researches that were made on various multipliers with different algorithm and achieved performance of the multipliers. The proposed system that achieved a better performance results comparing to the existing design. While it achieved improved area of 31%, power of 89.35% and delay of 11.98% for 4x4 multiplier and also achieved better results for high order multipliers and achieved improved area of 33.6%, power of 69.86% and reduced computational time of 10.93%. hence the proposed multipliers can be used in image processing applications where the signal-to-noise ratio (PSNR) of the image quality measured is compared with the accurate multipliers and the results obtained illustrates that our proposed multiplier operates better than the previous approximate multipliers.
Table 3. Comparison table and performance analysis of various multipliers with proposed multiplier

|          | AREA   | POWER  | DELAY  | ADP    | PDP    |
|----------|--------|--------|--------|--------|--------|
| 4x4      |        |        |        |        |        |
| Array    | 118.33 | 2835.93| 12.37  | 146267 | 35053  |
| Baugh wooley | 131.52 | 2991.97| 14.48  | 190296 | 43294  |
| Wallace  | 134.83 | 3055.23| 12.26  | 165177 | 37425  |
| Dadda    | 115.74 | 2782.35| 12.44  | 143853 | 34585  |
| Existing Vedic multiplier | 330   | 12514.94| 12.63  | 108405 | 26034  |
| Proposed Vedic multiplier | 310   | 11183.08| 11.98  | 1857.8 | 66986  |
| 8x8      |        |        |        |        |        |
| Array    | 544.63 | 13056.48| 36.29  | 19.758 | 473688 |
| Baugh wooley | 605.95 | 13863.21| 37.83  | 22917  | 524307 |
| Wallace  | 619.27 | 14154.46| 35.62  | 22052  | 504041 |
| Dadda    | 534.18 | 12708.37| 36.88  | 19706  | 468812 |
| Existing Vedic multiplier | 598.53 | 14373.29| 20.93  | 12521  | 300688 |
| Proposed Vedic multiplier | 1259 | 40683.82| 18.62  | 7657.6 | 367431 |

4.1 Comparison chart of various multipliers

Figure 7. Area comparison graph of various multipliers

From the above chart we can observe the performance analysis of various multipliers and the proposed multipliers can be achieved the optimized performance parameters comparing to other existing multipliers such as area.

Figure 8. Power comparison graph of various multipliers
The above chart gives a clear conclusion about the optimized power performance of proposed Vedic multiplier using reversible logic gates than all other existing Vedic multipliers.

![Image](delay.png)

**Figure 9.** Delay comparison chart of various multipliers

From the above chart gives a clear conclusion about the optimized delay performance of proposed Vedic multiplier using reversible logic gates than all other existing Vedic multiplier this is due to decrease in combination delay of 2nd stage of multipliers algorithm. In contrast to this increases the area delay product and power delay product of the design.

### 4.2 FPGA implementation of Vedic multiplier

The proposed improved design and existing available multipliers are synthesized and simulated using VHDL Coding style and it is implemented using cadence virtuoso EDA tool. The proposed multipliers are implemented in Spartan-6 FPGA family devices. The performance of the FPGA Device family Spartan-6 is analyzed by combinational Path delay of the multipliers. The proposed 4x4 Vedic multiplier achieved reduced combinational path delay of about 9.11% while implementing in FPGA device family. The below table shows the improved delay performance of various multiplier.

| Multipliers         | Spartan-6 |
|---------------------|-----------|
| Array-Multiplier    | 11.388    |
| Baugh-Wooley Multiplier | 12.3     |
| Wallace-tree Multiplier | 12.11    |
| Dadda Multiplier    | 12.15     |
| Vedic Multiplier    | 9.53      |
| Existing Vedic Multiplier | 9.503    |
| Proposed Vedic Multiplier | 9.112   |

### 4.3 Image Processing Application

To analyze the performance analysis of multiplier which is designed using 3-1-1-2 compressor Vedic multiplier using reversible logic gate. Image contrasting is implemented with proposed Vedic multiplier. MATLAB tool is used to simulate the image. For this we have choose a color images of size 512x512 having RGB pixels. Therefore, the available count of pixels is 786432 and pixel holding value of 8bits. Each pixel holds the range of (0-255). The size of our multiplier is also 8x8 bit therefore it is not necessary to convert pixels. All input pixels are mapped from the value (0-255) to (0-65535).

Contrasting of image is done using equation given below

\[ A = (\text{Input Image} \times \pi) \div 65535 \]
Contrast Image = (1 − cos(A) ÷ 2)

Quality of image is measured based on peak signal noise ratio (PSNR) of the output image is computed based on mean squared error (MSE). The PSNR value above 30 is considered as good quality of image. Input Image simulated also holding the PSNR value above 30 considered to be good quality image which is outcome of 8x8 proposed Vedic multiplier.

\[
MSE = (1 ÷ mp) × (\sum_{x=0}^{x=n-1} \sum_{y=0}^{y=m-1} (x(i,j) − y(i,j))^2)
\]

\[
PSNR = 10 \log_{10} \times \left( \frac{MAX^2 \times MSE}{MSE} \right)
\]

5.0 Conclusion

In this article, we proposed a compressor 3-1-1-2 using reversible logic which is a field that mainly concentrating to reduce the power, area and the combinational delay in summation part of partial products. This proposed design achieved reduced value of surplus power consumption upto 11.98% and limited combination delay upto 5.28%. The power, delay, and area of any device is always being an uncompromising parameter while designing any circuit in VLSI field. This approach showed a contrasting difference between the existing and proposed modified design which reduced the functioning logic gate count of 31%, during data computation for 4x4 multiplier and 33.6% for 8x8 multiplier. Therefore 3-1-1-2 compressor-based result can be most widely applied in real time applications such as Filters, cryptography, DSP processors, multimedia applications etc.

References

[1] Chen O, Sheen R and Wang S 2000 A low-power adder operating on effective dynamic data ranges IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 10 435–453
[2] Kung S R and Wang J P 2010 Design of power-efficient configurable booth multiplier IEEE Trans. Circuits. Syst.–I: Regular Papers 57 568–580
[3] Wang J P, Kuang S R and Liang S C 2011 High-accuracy fixed-width modified booth multipliers for lossy applications IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 19 52–60. 583
[4] Cilardo A, Caro D D, Petra N, Caserta F, Mazzocca N, Napoli E and Strollo A G M 2014 High
speed speculative multipliers based on speculative carry-save tree IEEE Trans. Circuits Syst.–I: Regular Papers. 61 3426–3435
[5] Sivanandam K and Kumar P 2015 Run time reconfigurable modified Vedic multiplier for high speed multimedia applications IEEE Int. Conf. Intell. Syst. Control, 2109–2113
[6] Wen M C, Wang S J and Lin Y N 2005 Low-power parallel multiplier with column by passing Electron Lett. 41 581–583
[7] Chang C H, Gu J and Zhang M 2004 Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits IEEE Trans. Circuits Syst.-I 51 1985–1997.
[8] Ramakrishnan M and Mallick P S 2017 Design and analysis of multiplier using approximate 15-4 compressor IEEE Access 5 1027–1036
[9] Veeramachaneni S, Avinash L, Puppala S R and Srinivas M B 2007 Novel architecture for high-speed and low-power 3-2, 4-2 and 5-2 compressors Inter- national Conference on VLSI design (VLSID’07) 324-329
[10] Kuang S R and Wang J P, 2010 Design of power-ecient configurable booth multiplier IEEE Trans. Circuits Syst.-I: Regular Papers 57(3) 568–580
[11] Kuang R S, Wang J P and Cang-Yuan 2009 Modified booth multiplier with a regular partial product array IEEE Trans. Circuits Syst.-II 56 404–408
[12] Qian Z and Margala M 2016 Low-power split-Radix FFT processors using Radix-2 Butterfly units IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 24 3008–3012
[13] Lamberti F, Andrikos N, Antelo E and Montuschi P 2011 Reducing the computation time in (short bit-width) two’s complement multipliers IEEE Trans. Comput. 60 148–156
[14] Lai S C, Juang W H, Lee Y S, Chen S H, Chen K H, Tsai C C and Lee C H 2016 Hybrid architecture design for calculating variable-length Fourier transform IEEE Trans. Circuits Syst.-II 63 279–283