Implementation of Low Power High Speed 16-Bit Approximate Multiplier for DSP Applications

M. Mahesh
Assistant Professor, Sreenidhi Institute of Science and Technology

Abstract: In many of the applications or packages, we do not require accurate results, like records processing and virtual signal processing that is message or data are packed together to achieve the desired result. Therefore, by using the layout of multiplier, we can consume speed of the process and power used by the processor, but multiplier effects a large area on the delay and energy intake on mathematical functions especially on the arithmetic processor. The parameters like speed and power can be covered by the use of an approximate multiplier. This paper has been designed for a 16-bit approximate multiplier. The complexities of the addition of those partial distributions are reduced based on the possibility. The proposed multiplier gains greater velocity and power consumption in comparison to the previous particular multiplier have seen in the synthesis results.

Keywords: Accurate process, compressors, product multiplier

I. INTRODUCTION

The logic circuit requires additional circuits like adders and multipliers to operate an arithmetic gadget. The performance of a processor depends on the energy dissipation and delay the results of the circuit. By using approximate mathematics circuits, hardware complexity will be reduced, but it comes along with losses of accuracy on power and velocity in digital structures [1-3]. Since most approximate designs select simplified good mechanism, that has a reduced power intake. To have accurate results in the circuits we choose multiplication operation, one of the arithmetic functions. In DSP algorithms multiplication is the basic operation. By using, approximate computing one can get inexact results. In the seek engine we can see these scenarios where no actual result may also exist for a selected search query. The addition of a small number of products, which intake greater energy, is one of the strategies of approximation.

By the rapid growth of performances, a small number of products are altered to introduce terms with specific possibilities. By the use of systematic approximation, altered partial merchandise is located in probability statistics. In many of the mathematics, circuits that perform specific operations are facing difficulties in overall performance development. In 1965, Luigi Dadda evolved a new technique on multiplier called Dadda multiplier. It is similar to the Wallace tree multiplier these techniques require very little place and produce results very quickly. It can be listed in 3 steps that is by multiplying each bit of one argument with each bit of other arguments until all arguments are increased.

- It reduces the generated partial products into layers
- The generated two values are arranged and collected them with the help of an adder.

In these, we designed an 8*8 Dadda multiplier. We introduce compressors instead of using simple full adders and half adders for the layout of a multiplier, which helps to compress the area, and for less energy consumption as compared with the previous adders, which are used within the multiplication method.

The main contributions and organization of this paper are summarized as follows: In section 2 we describe background details of approximate multiplier. Section 3 discusses the proposed work. Section 4 deliberates Results and Discussions. Finally, in section 5, we concluded the papers

II. BACKGROUND WORKS

Approximation in generating partial products: The partial products uses an approximate 2×2-bit multiplier design to perform a function by using an under-designed multiplier (UDM). By multiplication of "11" and "11" we acquire with a result "111" to save one output bit, but when in comparison with an approximate multiplier which results in "1001". Considering each input bit is similar, the 2×2-bit multiplier is used to design higher bit multipliers. Error is allowed when generating partial products in the multiplier.
Approximation in the partial product tree: To have high-speed addition of the partial products, we use a broken array multiplier (BAM). The carry-save adders operate on high-speed operations of BAM, which are used in array multipliers in both the directions. By the use the error-tolerant multiplier (ETM) which split multiplication into two types as the MSB's and LSB's. To operate these types of conditions a NOR gate-based control block is required:

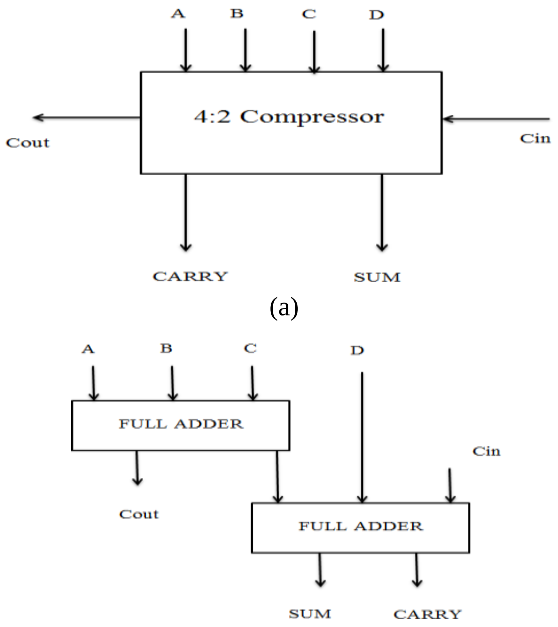
i) If zero is the result of MSBs, then at that point the augmentation segments the LSBs will be duplicated without any approximation, and

ii) If one is the result of MSBs, then the augmentation segment is enacted to duplicate the MSBs and LSBs will be utilized by the non-increase area by inexact multiplier process.

By the use of Wallace tree multiplier (WTM), plan a power and territory productive will surmised [4]. An *n*-bit WTM is activated by four *n*/2-bit sub-multipliers, and the hugest *n*/2-bit sub-multiplier is additionally executed by four *n*/4-bit sub-multipliers. The WTM is arranged into four unique modes by the quantity of surmised *n*/4-bit sub-multipliers in the hugest *n*/2-bit sub-multiplier. The estimated halfway items are then gathered together by a Wallace tree.

Partial Product Accumulation: The addition of the partial products can be done by using approximate adders, but by the usage of those adders, there will be less chance of accuracy in the process. The newly proposed approximate adders can gain excessive pace and less power consumption in the circuits [5-7]. The important factors are reduced and the delay of the adders will be very much lesser as compared to the previous adders and to associate the statistics in parallel form, a brand-new *n*-bit adder has been implemented.

4 to 2 Compressor Design: There are 5 inputs in 4-2 Compressor layout that are denoted by *A*, *B*, *C*, *D*, and C_{in} to get three outputs as Sum, Carry and Carry$_{\text{out}}$. Where the four inputs such as *A*, *B*, *C* and *D*, and the output Sum are having the same weight, while the input C_{in} is the output from a previous least enormous bit compressor and the C_{out} output is for the compressor within the subsequent level as shown in Figure 1.



(a)

(b)

Figure 1: (a) 4:2 Adder Compressor (b) 4:2 Adder Compressor Implemented with Full Adders

III. METHODS AND METHODOLOGY

The main aim of the multiplier is to focus on a particular area on the reduction of the partial product switch, which can additionally lessen the region, delay and strength consumption of the multiplier. To obtain the higher overall performance the adders are changed with the compressors in the approximate multiplier. By the usage of approximate compressors, which opposed to using the exact compressors, the complexity of the circuits on the multiplier is reduced. The implementation of the proposed approximate multiplier involves the half adder or complete adder which has grouped to the following discount level. The MSB and LSB bits used inside the approximation element and truncation part and those are brought within the next discount levels. The resultant bits are applied to the partial product discount tree and these are used inside the subsequent addition system. Hence, a simplified multiplier was designed with less variety of adders and which can produce the outputs at a very high speed.

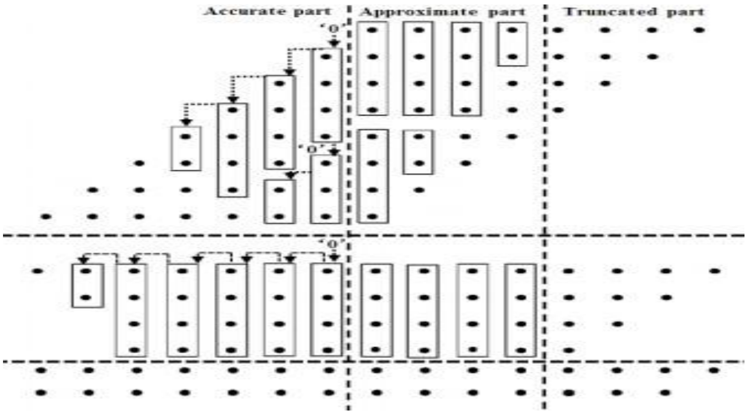


Figure 2: Proposed approximate compressors for a multiplier

The usage of truncation and the proposed approximate compressors for a multiplier in partial product reduction as in Figure 2. We can obtain a result with less energy intake and along with decreased hardware by the usage of the approximate part and the truncation element. The bits will lessen the loss of accuracy with the correct usage of compressors. The layout of the multiplier includes three ranges in those tiers the generation of the partial merchandise is the primary degree and the reduction of the partial merchandise will take place in the 2nd degree and inside the 1/3 degree, the final addition is accomplished. With the using the approximate compressors, strength intake will be reduced and the generation of the partial products along with the addition of the partial products may be finished efficaciously. The possibility of the partial products $a_{m,n}$ is obtained from the static factor of view in the proposed multiplier. If there is a wider variety of partial products $a_{m,n}$ and $a_{n,m}$ are mixed to shape propagate and generate signals. The partial merchandise is obtained from the altered partial products are $p_{m,n}$ and $g_{m,n}$. With the change in general and propagate signals the partial merchandise $a_{m,n}$ and $a_{n,m}$ might be generated from the altered partial products.

$p_{m,n}= a_{m,n}+ a_{n,m}$

$g_{m,n}= a_{m,n}\cdot a_{n,m}$

They generate alerts from the altered partial products having the possibility of one is being 1/16, that's decrease than the chance of the partial merchandise generated using the $a_{m,n}$. The chance of the am, n of being one is ¼. Hence, the partial merchandise obtained from the altered partial products gain less energy intake. When we are making use of the approximation to the partial products, they can acquire a higher overall performance. In the partial manufacturing discount tree, the OR gates are used in the accumulation stages and these can generate and propagate the outputs with the possibility of errors. The chance of mistakes is obtained by using the OR gates and these are used for the discount of generating and propagate alerts. When the number of propagating indicators is increasing the chance of mistakes also increases linearly. Hence, the value of the error additionally will increase. To lessen the opportunity of mistakes the maximum number of bits are propagated the usage of or gates consequently they generate signals that are reduced. The partial products are gathered with the possibility of generating and propagate indicators that are acquired from the altered partial products. In the build-up level the approximate 1/2 adder, complete adder, and 4:2 compressors are used. With the usage of those approximate adders, the bring bits will propagate faster and the approximate adders will generate outputs. Hence the sum and convey bits are once more accumulated to the following reduction level alongside the truncated bits. With the reduction of the partial products, the opportunity of error additionally decreased. The sum and convey bits are propagated by way of using the following equations

Sum = $x_1 + x_2$

Carry = $x_1 \cdot x_2$.

The approximate full adder the XOR gates are replaced with or gates to generate the sum. With the change of the whole adder operation, there is an error incidence inside the last levels; this produces the difference among the authentic and approximate values.

$W = (x_1 + x_2)$

Sum = $W\oplus x_3$

Carry = $W \cdot x_3$.

In the proposed approximate compressor design, there will be 4 inputs and it will produce 3 outputs. Here the 3 outputs are one in handiest one circumstance out of all viable conditions. The minimum error can be removed by the distinction, which is calculated, and its miles have given as one for the closing one feasible condition. Hence, in the following equation sum computation will be modified.

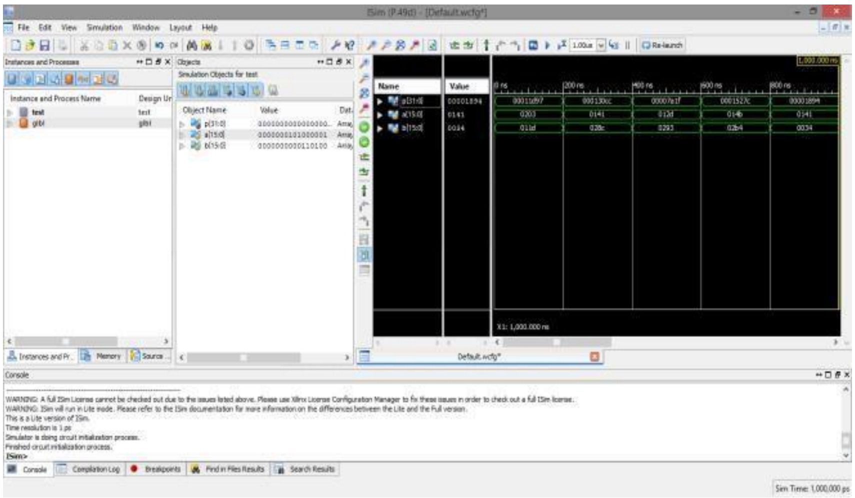
$W_1 = x_1\cdot x_2$

$W_2 = x_3\cdot x_4$

Sum = $(x_1 \oplus x_2) + (x_3 \oplus x_4) + W_1 \cdot W_2$
Carry = $W_1 + W_2$.

IV. RESULTS AND DISCUSSION

By the use of Verilog HDL language and synthesized by using the XILINX tool we designed a 16-bit approximate multiplier. The Xilinx provides a clear area, feature, and dynamic and static power intake. For the result, the generated partial products are accelerated and compressed by using an approximate half adder, full adder, and 4:2 compressor designs. The proposed multiplier achieves better performance as compared with the previous approximate multipliers. In less power consumption and use minimum area for the process.



Object Name	Value
p[31:0]	0000000000000000...
a[15:0]	0000000101000001
b[15:0]	0000000000110100

```
WARNING: A full ISim License cannot be checked out due to the issues listed above. Please use Xilinx License Configuration Manager to fix these issues in order to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Figure 3: 16-bit approximate Multiplier simulation result

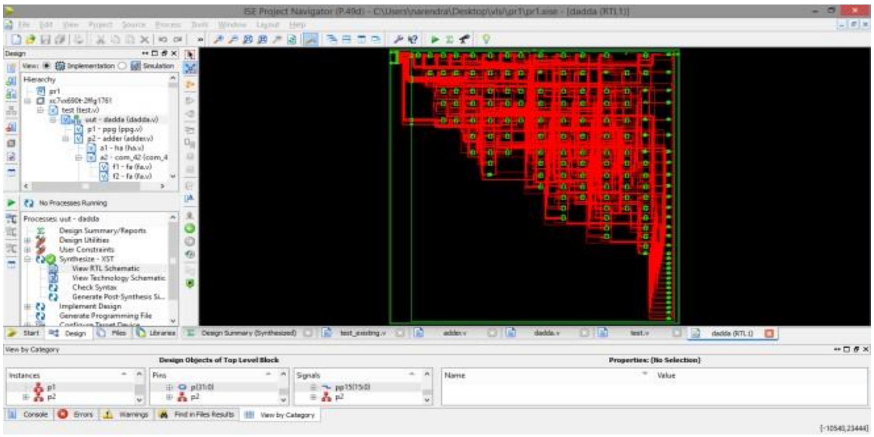


Figure 4: 16-bit approximate multiplier RTL schematic

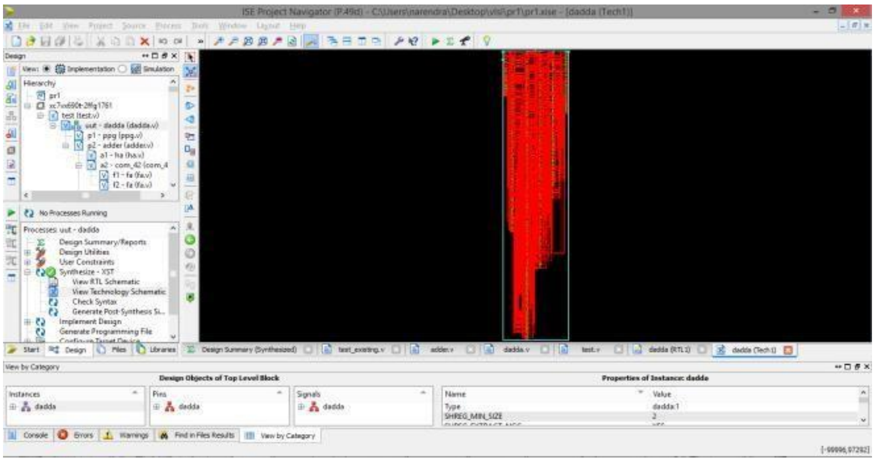
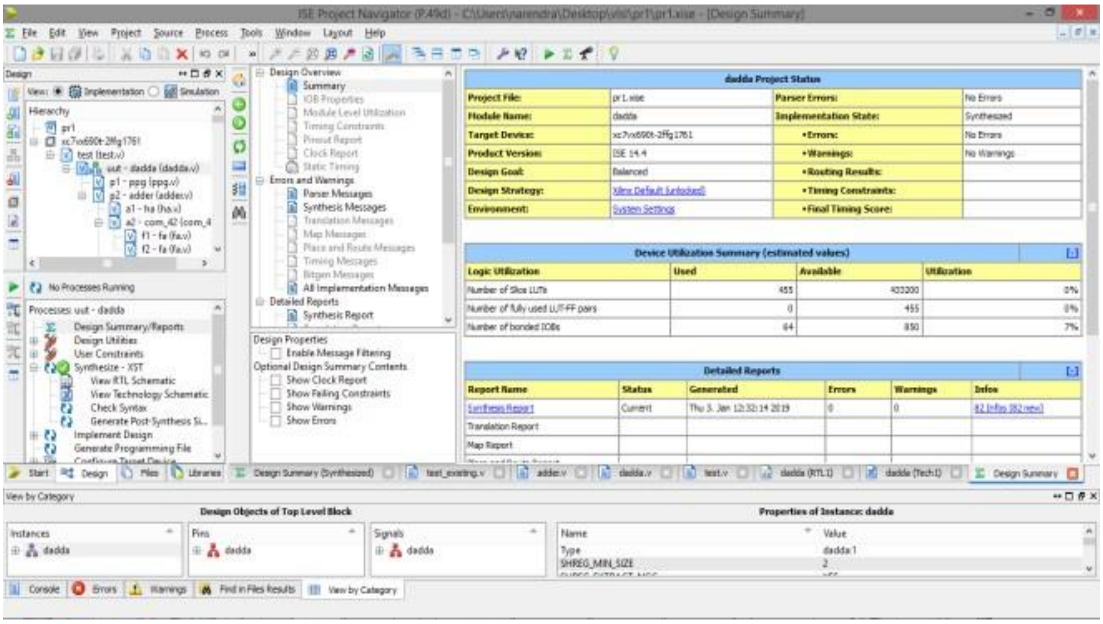


Figure 5: 16-bit approximate multiplier technology schematic.



dadda Project Status			
Project File:	pr1.xise	Parser Errors:	No Errors
Module Name:	dadda	Implementation State:	Synthesized
Target Device:	xc7vx690t-2ffg1761	•Errors:	No Errors
Product Version:	ISE 14.4	•Warnings:	No Warnings
Design Goal:	Balanced	•Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	
Environment:	System Settings	•Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	455	433200	0%
Number of fully used LUT-FF pairs	0	455	0%
Number of bonded IOBs	64	850	7%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Thu 3. Jan 12:32:14 2019	0	0	82 Infos (82 new)
Translation Report					
Map Report					

Properties of Instance: dadda

Name	Value
Type	dadda:1
SHREG_MIN_SIZE	2

Figure 6: 16-bit approximate multiplier summary report

V. CONCLUSION

The proposed 16-bit approximate multiplier designed by the use of the altered partial merchandise. By the use of approximate adders in the partial production discount tree, we came to reduce the altered partial merchandise and decrease the partial merchandise products. The higher pace in comparison is achieved by using the approximate half adder, full adder, and 4:2 compressor compared to the previous multipliers. By using these, better qualities can be achieved while in comparison to the previous approximate multiplier designs. The given or explained multipliers can extensively be used within the packages or algorithm with minute losses in output along with less power consumption and location.

REFERENCES

[1]. J. Han and M. Orshansky. "Approximate Computing: An Emerging Paradigm for Energy-Efficient Design." In approaches of the eighteenth IEEE European Test Symposium, Avignon, France, May 2013, pp.1-6.

[2]. J. Liang, J. Han and F. Lombardi." New measurements for the unwavering best of envisioned and probabilistic adversities Trans. PCs. Vol. Sixty-two, no.9, pp.1760-1771, Sept. 2013.

[3]. Momeni, J. Han, P. Montuschi and F. Lombardi. "Structure and Analysis of Approximate Compressors for Multiplication." IEEE Trans. PCs, vol.64, no.4, pp. 984-994, Apr. 2015.

[4]. C.H. Lin and I.C. Lin. "High exactness estimated multiplier with blunder treatment." In Proc. ICCD 'thirteen: In the 2013 IEEE thirty first International Conference on Computer Design (ICCD). Asheville, NC, USA, Oct. 2013, pp. 33-38.

[5]. K. Bhardwaj, P.S. Maneand J. Henkel. "Power-and territory talented Approximate Wallace Tree Multiplier for blunder versatile frameworks." In Sym. ISQED'14: In fifteenth International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA. Blemish. 2014, pp. 263-269.

[6]. S. NarayanaMoorthy, H.A. Moghaddam, Z. Liu, T. Park, N.S. Kim, "Vitality Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications." IEEE Trans. Large Scale Integration Systems (VLSI), vol.23, no.6, pp.1180-1184, Jun. 2015.

[7]. M.S. Lau, K.V. Ling and Y.C. Chu. "Vitality conscious probabilistic multiplier: shape and exam." In Proceedings of the 2009 typical assembly on Compilers, engineering, and amalgamation for inserted frameworks, Grenoble, France, Oct. 2009, pp.281-290.