ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1s/309-317
P. Chatrika *et. al.*, / International Journal of Engineering & Science Research

Design And Implementation Of ROBA Multiplier In DSP Systems

[1]Mrs. J. Stella Mary, [2]P. Chatrika, [3]K. Harini, [4]B. Kusuma
[1]Associate professor, Electronics and Communication Engineering, BRECW
[2,3,4]B.Tech Students, Department of Electronics and Communication Engineering, BRECW

ABSTRACT
In this project, we will be discussing about ROBA(Rounding Based Approximate) multiplier that is high speed yet energy efficient. The approach is to round the operands to the nearest exponent of two. This way the computational intensive part of the multiplication is omitted improving speed and energy consumption at the price of a small error. The proposed approach is applicable to both signed and unsigned multiplications. We discuss three hardware implementations of the approximate multiplier that includes one for the unsigned and two for the signed operations. The efficiency of the proposed multiplier is evaluated by comparing its performance with those of some approximate and accurate multipliers using different design parameters. In addition, the efficiency of the proposed approximate multiplier is studied in two image processing applications, i.e., image sharpening and smoothing.

1-Introduction

Multipliers are one of the most significant blocks in computer arithmetic and are generally used in different digital signal processors. There is growing demands for high-speed multipliers in different applications of computing systems, such as computer graphics, scientific calculation, and image processing and so on. Speed of multiplier determines how fast the processors will run and designers are now more focused on high speed with low power consumption. The multiplier architecture consists of a partial product generation stage, partial product reduction stage and the final addition stage. The partial product reduction stage is responsible for a significant portion of the total multiplication delay, power and area. Therefore in order to accumulate partial products, compressors usually implement this stage because they contribute to the reduction of the partial products and also contribute to reduce the critical path which is important to maintain the circuit's performance.
Addition and multiplication are widely used operations in computer arithmetic; for addition full-adder cells have been extensively analyzed for approximate computing Liang et al. has compared these adders and proposed several new metrics for evaluating approximate and probabilistic adders with respect to unified figures of merit for design assessment for inexact computing applications. For each input to a circuit, the error distance (ED) is defined as the arithmetic distance between an erroneous output and the correct one. The mean error distance (MED) and normalized error distance (NED) are proposed by considering the averaging effect of multiple inputs and the normalization of multiple-bit adders.
However, the design of approximate multipliers has received less attention. Multiplication can be thought as the repeated sum of partial products; however, the straightforward application of approximate adders when designing an approximate multiplier is not viable, because it would be very inefficient in terms of precision, hardware complexity and other performance metrics. Several approximate multipliers have been proposed. Most of these designs use a truncated

multiplication method; they estimate the least significant columns of the partial products as a constant. An imprecise array multiplier is used for neural network applications by omitting some of the least significant bits in the partial products (and thus removing some adders in the array). A truncated multiplier with a correction constant is proposed.

2-LITERATURE SURVEY

1. "Design of Low-Power ROBA Multiplier" (2020) - IEEE Transactions

Approximation Techniques: Strategies for rounding operands before multiplication, reducing power and area usage.

Power Efficiency: Emphasizing reduced power consumption for portable and embedded systems.

Performance Evaluation: Comparing ROBA multipliers to both precise and other approximate multipliers, particularly in image processing applications like smoothing and sharpening.

These techniques are particularly relevant for error-tolerant applications.

2. "ROBA Multiplier for DSP Applications" (2019) - Journal of Signal Processing

Rounding-Based Approximation: The operands are rounded to powers of two to simplify multiplication and improve speed while maintaining energy efficiency.

Error-Tolerant Applications: The design targets systems like image processing or voice smoothing, where minor computational errors can be tolerated.

High-Speed and Low-Power Design: The architecture reduces both area and power consumption without significant accuracy loss.

These optimizations make it ideal for DSP and multimedia applications.

3. "Area-Efficient ROBA Multiplier" (2018) - ACM Transactions on Embedded Computing Systems

Energy Efficiency: Reducing power consumption through approximate arithmetic, particularly for error-resilient applications.

High-Speed Operation: Accelerating computation by approximating multiplication operations.

Application in Image Processing: ROBA multipliers are applied to tasks like image sharpening and smoothing while balancing accuracy and performance(SpringerLink).

4. "Design of Approximate Multipliers" (2019)

The book Design of Approximate Multipliers (2019) discusses various topics in approximate computing, with a focus on energy-efficient multiplier designs. The ROBA (Rounding-Based Approximate) multiplier is central to this, aiming to reduce power consumption and complexity by rounding input values to powers of two before performing multiplication. Topics covered include energy-efficient algorithms, error analysis, and the application of these multipliers in areas like digital signal processing and image processing, where minor inaccuracies can be tolerated for significant gains in speed and power efficiency(SpringerLink).

3-ROBA MULTIPLIER IN DSP SYSTEMS

ENERGY minimization is one of the main design requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets. It is highly desired to achieve this minimization with minimal

performance (speed) penalty. Digital signal processing (DSP) blocks are key components of these portable devices for realizing various multimedia applications. The computational core of these blocks is the arithmetic logic unit where multiplications have the greatest share among all arithmetic operations performed in these DSP systems. Therefore, improving the speed and power/energy-efficiency characteristics of multipliers plays a key role in improving the efficiency of processors. Many of the DSP cores implement image and video processing algorithms where final outputs are either images or videos prepared for human consumptions. This fact enables us to use approximations for improving the speed/energy efficiency. This originates from the limited perceptual abilities of human beings in observing an image or a video.

VLSI DESIGN:

VLSI stands for Very Large-Scale Integration. Very Large-Scale Integration is the process of creating integrated circuits by incorporating thousands of transistor circuits into a single chip. This trend is continuing with very important implications on Very Large-Scale Integration and systems design. One of the important applications of information services is their increasing need for very high processing power and bandwidth.The other important application is that the information services tend to become more and more personalized as opposed to collective services such as broadcasting. Very Large-Scale Integration began in the 1970s. The microprocessor is a Very Large-Scale Integration device. This field contains packing more number of logic devices into smaller areas.

Existing System

Multipliers play an important role in today's digital signal processing and various other applications. Essential design targets of multiplier include high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier are required thereby making them suitable for various VLSI implementations. The straightforward way to implement a multiplication is based on an iterative adder-accumulator for the generated partial products as shown in figure 3.1. This multiplier is called a serial multiplier. However, this solution is quite slow as the final result is only available after 'n' clock cycles, 'n' is the size of the operands. Serial multipliers are used where area and power are of utmost importance and increased delay can be tolerated. A faster version of the iterative multiplier should add partial products at once. This could be achieved by unfolding the iterative multiplier and yielding a combinational circuit that consists of several partial product generators together with several adders that operate in parallel. This multiplier is called Parallel Multiplier and is shown in figure 3.2.

25

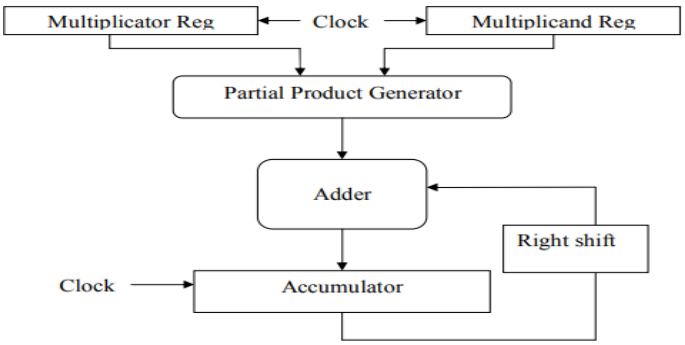

Figure 2.1 Iterative multiplier

Proposed System

In addition to the image and video processing applications, there are other areas where the exactness of the arithmetic operations is not critical to the functionality of the system. Being able to use the approximate computing provides the designer with the ability of making tradeoffs between the accuracy and the speed as well as power/energy consumption. Applying the approximation to the arithmetic units can be performed at different design abstraction levels including circuit, logic, and architecture levels, as well as algorithm and software layers. The approximation may be performed using different techniques such as allowing some timing violations (e.g., voltage over scaling or over clocking) and function approximation methods (e.g., modifying the Boolean function of a circuit) or a combination of them. In the category of function approximation methods, a number of approximating arithmetic building blocks, such as adders and multipliers, at different design levels have been suggested. In this paper, we focus on proposing a high-speed low power/energy yet approximate multiplier appropriate for error resilient DSP applications. The proposed approximate multiplier, which

is also area efficient, is constructed by modifying the conventional multiplication approach at the algorithm level assuming rounded input values. We call this rounding-based approximate (RoBA) multiplier.

Block Diagram

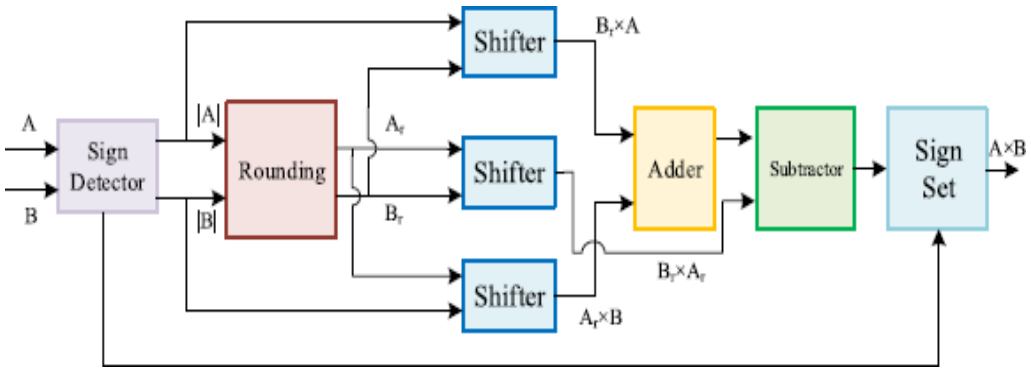

Figure 2.4 Block diagram for the hardware implementation of the ROBA multiplier

4-Software Requirements

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

Software Requirements

For developing the application following are the Software Requirements:

Xilinx software tools are used for designing, simulating, and implementing digital circuits on Xilinx FPGAs and SoCs. The two main software platforms are:

Vivado Design Suite:

Purpose: For FPGA and SoC design, including synthesis, simulation, and hardware implementation. Features: High-level synthesis (HLS), IP integration, timing analysis, and debugging.

Use: Supports newer Xilinx FPGAs like the 7-series, Zynq SoCs, and Ultra Scale devices.

IP Modules:

If your design includes IP modules that were created using CORE Generator™ software or Xilinx® Platform Studio (XPS) and you need to modify these modules, you may be required to update the core. However, if the core net list is present and you do not need to modify the core, updates are not required and the existing net list is used during implementation.

5-ADVANTAGES, DISADVANTAGES AND APPLICATIONS

Advantages

1. **High Speed**

Reduced Latency: The ROBA (Reduced Operand Binary Arithmetic) multiplier architecture is designed to minimize the number of adders and intermediate steps, resulting in faster computation times compared to traditional multipliers.

2. **Low Power Consumption**

Efficient Resource Utilization: The ROBA multiplier typically consumes less power due to its efficient use of resources, making it suitable for battery-operated devices and applications where power efficiency is critical.

3. **Area Efficiency**

Compact Design: The design often requires fewer gates and resources compared to other multiplication methods, leading to a smaller chip area. This is particularly beneficial for integrating multiple DSP functions in a single chip.

Disadvantages

Design Overhead

Additional Resources: Although it is area-efficient, the initial design and testing phases might require additional time and resources to ensure proper implementation and verification of the ROBA multiplier.

Higher Development Cost

Investment in Design Tools: The use of advanced design methodologies and tools (like Vivado or other EDA tools) may increase development costs, especially for small projects or startups with limited budgets

Applications:

1. **Digital Signal Processing (DSP)**

Audio and Speech Processing: The ROBA multiplier can be used in applications such as audio encoding/decoding, noise reduction, and speech recognition, enhancing the

performance of algorithms used in these areas.

2. Image and Video Processing

Image Filtering and Enhancement: It can be applied in image processing tasks such as convolution, edge detection, and image scaling, where efficient multiplication is critical for processing pixel data. Video Compression: Used in video codecs for tasks like motion estimation and video encoding, improving compression efficiency and playback quality.

6-RESULTS AND DISCUSSION

In this chapter, we will discuss about the results of two inputs given to the ROBA multiplier with single output in DSP systems

Results of the ROBA Multiplier for the given inputs include the following:

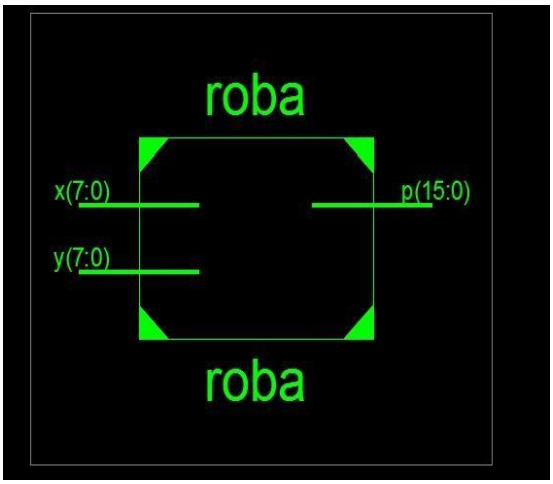

Figure 5.1: RTL schematic

The RTL (Register Transfer Level) schematic you've shared appears to be a part of a digital system, most likely representing a rounding-based approximate multiplier. While I can't access the details within the image directly, I can provide an explanation based on typical designs of such systems.

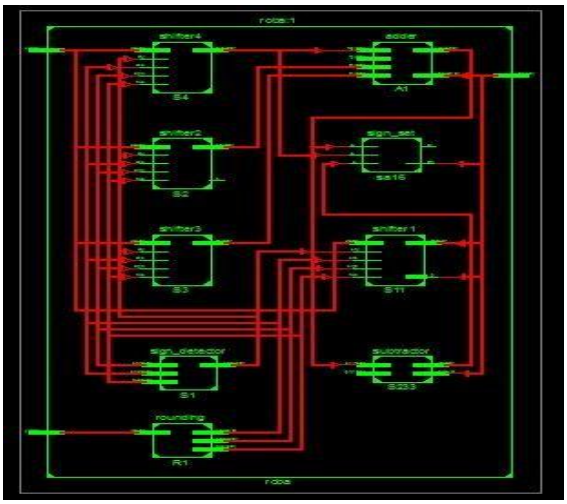

Figure 5.2: Internal block diagram

Data Flow:

1. Input operands are fed into the multipliers.
2. The partial products are shifted and aligned using the shifter units.
3. The adders then sum these partial products to form a preliminary result.
4. The rounding unit truncates or modifies the least significant bits, producing an approximate final output.

This design approach simplifies the traditional multiplication logic and is used in applications where the speed or power efficiency outweighs the need for full precision, such as in certain machine learning tasks, image processing, or low-power embedded systems.

Table : Area consumed by the ROBA multiplier for the given specific inputs

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	156	9,312	1%	
Number of occupied Slices	88	4,656	1%	
Number of Slices containing only related logic	88	88	100%	
Number of Slices containing unrelated logic	0	88	0%	
Total Number of 4 input LUTs	156	9,312	1%	
Number of bonded IOBs	32	232	13%	
Average Fanout of Non-Clock Nets	4.16			

The area consumed by the rounding-based approximate multiplier is quite low. It uses only 1% of the available LUTs and slices, while 13% of the IOBs are utilized for external connections. The design is efficient with no unrelated logic in the occupied slices. This makes it a resource-efficient implementation on the Spartan-3e FPGA.

```
    ------------------------------------
    Total                     21.412ns (13.455ns logic, 7.957ns route)
                                       (62.8% logic, 37.2% route)

=========================================================================


Total REAL time to Xst completion: 19.00 secs
Total CPU time to Xst completion: 19.02 secs
```

Figure 5.3: Delay

Logic Delay: 13.455 ns is spent on logic operations, which accounts for 62.8% of the total delay. This likely refers to the time taken for the combinational logic to compute the approximate multiplication.

Routing Delay: 7.957 ns is spent on routing, which is 37.2% of the total delay. This refers to the time needed to propagate signals between different

blocks (multipliers, adders, rounding unit) within the FPGA or digital circuit.

Additionally, the real time and CPU time for synthesis completion are both around 19 seconds, indicating the time taken by the synthesis tool to generate the design.

Total delay: 21.412 ns, with most of the time (62.8%) spent on logic operations and the remaining (37.2%) on routing.

Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Temp Grade	Commercial
Process	Typical
Speed Grade	-5

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	156	9312	2
Signals	0.000	146	--	--
IOs	0.000	32	232	14
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.081	0.000	0.081

Figure 5.4: Power

Power consumption of the rounding-based approximate multiplier is dominated by leakage power (0.081 W), with no dynamic power consumption indicated. Most of the power is drawn from the Vccint and Vccaux supplies. The junction temperature (27.1°C) is well within safe limits for operation under ambient conditions (25°C).

This suggests that the rounding-based approximate multiplier design is primarily power- efficient and consumes low dynamic power.

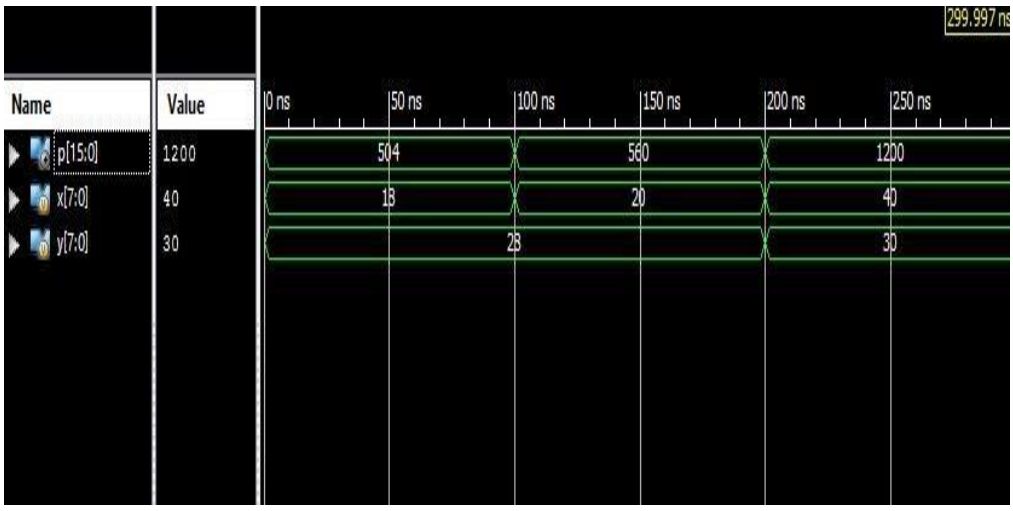

Figure 5.5: Simulation results

7-CONCLUSION

In this paper, we proposed a high-speed yet energy efficient approximate multiplier called RoBA multiplier. The proposed multiplier, which had high accuracy, was based on rounding of the inputs in the form of 2n. In this way, the computational intensive part of the multiplication was omitted improving speed and energy consumption at the price of a small error. The proposed approach was applicable to both signed and unsigned multiplications. Three hardware implementations of the approximate multiplier including one for the unsigned and two for the signed operations were discussed. The efficiencies of the proposed multipliers were evaluated by comparing them with those of some accurate and approximate multipliers using different design parameters. The results revealed that, in most (all) cases, the RoBA multiplier architectures outperformed the corresponding approximate (exact) multipliers.

This work is extended to implement fir filter using roba multiplier. It offers great advantage in the reduction of delay.

REFERENCES

1. Manchiryala Manogna, "ROBA Multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy-Efficient Digital Signal Processing", IEEE November 2021.

2. Gollapalli Veerendra, S Narasimhulu, "Design and Implementation of ROBA Multiplier in DSP Systems", IEEE Trnas, vol 13. (VLSI) Syst., pp.540-547, September 2021.

3. P. Ramya, " Design and Implementation of ROBA Multiplier In DSP Systems", Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 365-370, November 2019.

4. P.Lohray, S. Gali, S. Rangisetti and T. Nikoubiu, "Rounding Technique Analysis for Power- Area & Energy Efficiency Approximate Multiplier Design," IEEE 2019.

5. "Design of Approximate Multipliers" (2019) - Springer.

6. . "Energy-Efficient Approximate Multipliers" (2020) - IEEE Transactions on Very Large-Scale Integration (VLSI) Systems.

Books Referred Are:

1. Digital design with an introduction to the Verilog HDL by M . Morris Mano.

2. Basic VLSI design by Douglas , A. Pucknell and Kamran Eshraghian.

3. Digital Signal Processing by Nagoor Kani.

.