An Estimated Multiplier For Quick Energy-Efficient Digital Indication Dispensation

BASUVULA THRIVENI
M.Tech Student, Dept of ECE, Priyadarshini Institute of Technology & Science for Women, Chintalapudi, Tenali, A.P, India

A.SOMASEKHAR
Associate Professor, Dept of ECE, Priyadarshini Institute of Technology & Science for Women, Chintalapudi, Tenali, A.P, India

Abstract: **We propose a high-speed, energy-efficient approximation multiplier. The method is to round the coefficients to the nearest exponent of two. In this way, the abbreviated arithmetic part is omitted from the multiplication process to improve the speed and power consumption with a small error rate. The proposed approach applies to both signed and unsigned complications. We propose that three devices be implemented for the coarse multiplier, which includes one for unsigned and two for signed operations. The efficiency of the proposed multiplier is evaluated by comparing its performance with the performance of some approximate and accurate multipliers using different design criteria. In addition, the effectiveness of the proposed approximate multiplier is studied in two image processing applications, i.e. image sharpening and smoothing.**

Keywords: - **Accurate Multiplier; Accuracy; MAC Unit.**

INTRODUCTION

Multiples are one of the most important blocks in computer computing and are commonly used in various digital signal processors. There are increasing demands for high speed multipliers in various applications of computer systems, such as computer graphics, scientific calculations, image processing, etc. Multiplying speeds determines how fast processors run and designers are now focusing more on high speeds with lower power consumption [1]. The structure of the multiplier consists of the partial generation stage of the product, the partial reduction stage of the product and the final addition stage. The partial product reduction phase is responsible for a large part of the total slave retardation, power and area. Thus, for the partial product assembly, compressors usually implement this stage because it contributes to the reduction of the component products and also contributes to the reduction of the critical path that is important to maintain the circuit performance.

This is achieved through the use of 3-2, 4-2, 5-2 compressor structures. The 3-2 compressor circuit is also known as the complete hose. As these compressors are used more frequently in larger systems, the improved design will significantly contribute to the overall system performance. The internal structure of the compressors consists mainly of XOR-XNOR gates and servo transmissions. XOR-XNOR circuits are also building blocks in various circuits, such as arithmetic circuits, multipliers, compressors, parity meters, etc. Optimized design of XOR-XNOR gates can improve the performance of multiplexing circuits. In the current work, a new XOR-XNOR unit has been proposed and a 4-2 compressor has been implemented with this unit. The use of the proposed circuit in the fractional accumulator of the product reduces the transistor number as well as the power consumption [2].

Addition and multiplication are widely used in computer calculations; In addition, whole-length whole-cell cells have been analyzed for approximate computation (Liang et al. Compare these additions and suggest some new measures to evaluate approximate and probable additions in relation to standardized design evaluation earnings figures for inaccurate computer applications). For each input in the circuit defined as the calculation distance between the erroneous output and the correct output. Average error distance (MED) and measured error distance (NED) were proposed by considering the effect of averages of multiple inputs and normalization of multi-bit additions [3]. NED is approximately constant with the implementation size and is therefore useful in assessing the reliability of a particular design. The compromise between accuracy and robustness has also been quantified.

LITERATURE SURVEY

The use of approximate multipliers in image processing applications, resulting in lower power consumption, delay, and transistor count compared to the fine multiplexer design, has been discussed in the literature. A precisely configurable multiplexing architecture (ACMA) has been proposed for fault-tolerant systems. To increase productivity, ACMA used a technique called pregnancy prediction, which was based on pre-logical computation. Compared with precision hitting, the suggested rounding hitting reduced latency by nearly 50% by reducing the critical path. Also Hardwar et al. provide the approximate multiplier for Wallace Tree (AWTM). Once again, she pleaded with prospects to reduce the critical path. In this work, AWTM was used in a real-time modular image application that reduced approximately 40% and 30% of power and area, respectively, without any image quality loss compared to using a Wallace Tree Multiplier (WTM) micro-architecture.

Unsigned approximate multiplication and division operations based on an approximate logarithm of the coefficients. In suggested multiplication, the sum of approximate logarithms determines the result of the operation. Thus multiplication operations are simplified into a small number of transform and addition operations. A method has been proposed to increase the accuracy of the multiplication approach. This was based on the dissolution of the edges of the input. This method greatly improved the mean error at the expense of increasing the approximate multiplier devices approximately 2 times. The dynamic segmentation (DSM) method is introduced, which performs the multiplication operation on the m-bit segment of the first bit of the input parameter. The DRUM multiplier (infinite dynamic range multiplier) has been proposed, which defines the m-bit segment starting with a single forward bit of input parameters and less significant bits of clipped values in a single set.

PROPOSED DESIGN ARCHITECTURE

In addition to image and video processing applications, there are other areas where mathematical precision is not important for system functionality. The ability to use approximate computers gives the designer the ability to compromise between accuracy, speed as well as power / power consumption. Rounding of computational units can be applied to different levels of design abstraction, including levels of circuit, logic and architecture, as well as low algorithm and software [4]. Approach can be done using different techniques, such as allowing time constraints (e.g. overvoltage or over clocking), methods of function approach (e.g. modulating the logical function of a circuit) or a combination thereof. In the category of methods for approximating functions, a number of approximate calculation building blocks, such as additions and multiples, have been proposed at different design levels. In this paper, we focus on a high-speed-low power / power proposal, but a suitable multiplier approach for flexible fault-tolerant DSP applications. The proposed approximate multiplier, which is also regionally efficient, is generated by modifying the traditional algorithm-level multiplication approach, assuming rounded input values. We call this approximate multiplier based on approximation (RoBA).

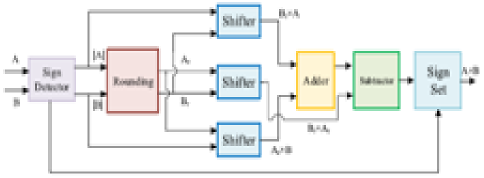


Fig.4.1 Block diagram for the hardware implementation of the ROBA multiplier.

A. *Algorithm for MROBA multiplier*

The main concept of the proposed fine multiplier on the basis of modified approximation is to design the multiplier so that it takes all values regardless of 2n. More detailed description of the multiplier [5]. Let us first consider Ar as the approximate input value of input A and Br as the rounded input value of input B. The multiplication of A * B is written as follows:

A * B = (Ar * B) (Br * B) + (Ar * B) + (Br * A) - (Ar * Br) - (1) ------ (A)The main key point to consider is the output of (Ar * B) and (Br * B) complex and the weight of the term will lead to small values compared to the exact numbers, so the product of this term can be omitted and this also leads to a complex approach to hardware design. Therefore, the multiplication with the following expression can be performed as A * B = (Ar * B) + (Br * A) - (Ar * Br) - [1] Product terms (Ar * B), (Br * A), (Ar * Br) can be implemented through three cylindrical commentator of N bits and one feed is executed by parallel prefix pebble gluing tool from N bits and one subtraction is also required. If the values of A or B are equal to N where (N = 1, 2, 3 ... N) have two rounded values N + 1 and N, depending on the inputs of A and B, the values are determined. The type of rounding values applies to both the condition a greater than b (a> b) and less than b (a <b). Except that for A is equal to B (A = B), the rounded entry for A and B would be the middle number for N + 1 and N, i.e. (2N + 1) / 2 in the traditional rounding-based exact multiplier, the numbers (which The proposal to enter ROBA) is seen in the form of 3x2p-2 (where p is a random positive number greater than one), and we also have two circled values in the form of 2p and 2p-1, both of which have the same effect, but in this case, the largest value It is the approximate value of both inputs, since larger values lead to smaller devices being implemented. But the accuracy of this multiplier is poor and the exact result is not obtained in this case. Therefore, we consider the ad hoc multiplier based on approximation which applies to all N numbers (N1, 2, 3 .... N) And the exact outputs of the selected input are obtained. Diagram 1.

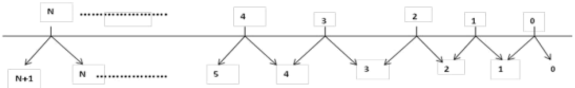


Fig 1: Top numbers represents inputs and below numbers represents the rounded values

Next block we have a basic adder we can add +1 i.e. (N+1). Now we need three shifters to compute the product terms (Ar *B), (Br * A), (Ar * Br). The amount of shifting is basically done on no of bits we considered and the shifting operation is performed on the rounded values of Ar and Br respectively Then the output of shifter 1 and shifter 3 are given has the input to the adder basically we considered a kogge-stone adder. And the output of the adder and the output of the shifter2 is given has the input to the Subtractor block and finally we have a sign set block which is needed if we considered the signed numbers. For the unsigned it is not necessary and the final multiplication A*B is obtained the output is exact for the given input when compared with the conventional rounding based accurate multiplier it gives the exact result for the input in all the possible conditions [6].

SIMULATION RESULTS

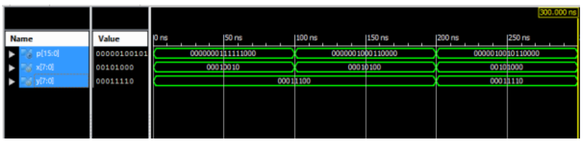


Fig 2: Simulation Waveform

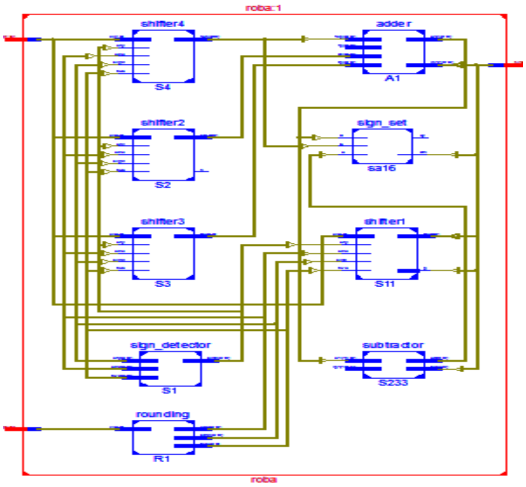
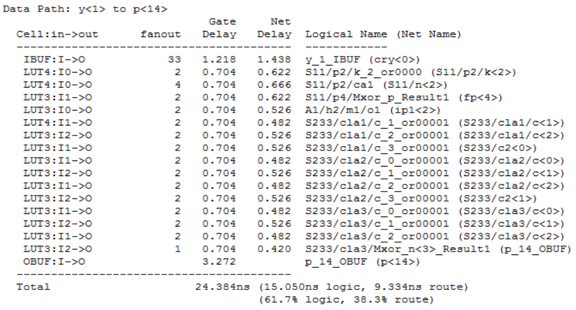
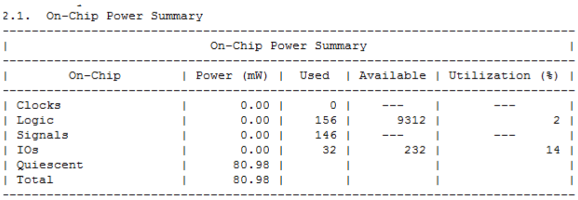


Fig 3: RTL Schematic



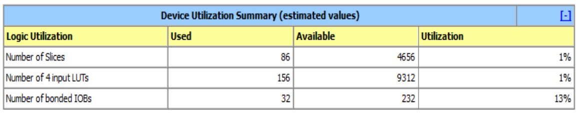
```
Data Path: y<1> to p<14>
                                 Gate     Net
  Cell:in->out       fanout    Delay    Delay   Logical Name (Net Name)
  ------------------------------------------   ------------
   IBUF:I->O              33    1.218    1.438  y_1_IBUF (cry<0>)
   LUT4:I0->O              2    0.704    0.622  S11/p2/k_2_or0000 (S11/p2/k<2>)
   LUT4:I0->O              4    0.704    0.666  S11/p2/ca1 (S11/n<2>)
   LUT3:I1->O              2    0.704    0.622  S11/p4/Mxor_p_Result1 (fp<4>)
   LUT3:I0->O              2    0.704    0.526  A1/h2/m1/o1 (ip1<2>)
   LUT4:I1->O              2    0.704    0.482  S233/cla1/c_1_or00001 (S233/cla1/c<1>)
   LUT3:I2->O              2    0.704    0.526  S233/cla1/c_2_or00001 (S233/cla1/c<2>)
   LUT3:I1->O              2    0.704    0.526  S233/cla1/c_3_or00001 (S233/c2<0>)
   LUT3:I1->O              2    0.704    0.482  S233/cla2/c_0_or00001 (S233/cla2/c<0>)
   LUT3:I2->O              2    0.704    0.526  S233/cla2/c_1_or00001 (S233/cla2/c<1>)
   LUT3:I1->O              2    0.704    0.482  S233/cla2/c_2_or00001 (S233/cla2/c<2>)
   LUT3:I2->O              2    0.704    0.526  S233/cla2/c_3_or00001 (S233/c2<1>)
   LUT3:I1->O              2    0.704    0.482  S233/cla3/c_0_or00001 (S233/cla3/c<0>)
   LUT3:I2->O              2    0.704    0.526  S233/cla3/c_1_or00001 (S233/cla3/c<1>)
   LUT3:I1->O              2    0.704    0.482  S233/cla3/c_2_or00001 (S233/cla3/c<2>)
   LUT3:I2->O              1    0.704    0.420  S233/cla3/Mxor_n<3>_Result1 (p_14_OBUF)
   OBUF:I->O                    3.272           p_14_OBUF (p<14>)
  ------------------------------------------
  Total                        24.384ns (15.050ns logic, 9.334ns route)
                                        (61.7% logic, 38.3% route)
```

Fig 4: Timing Report



```
2.1.  On-Chip Power Summary
-----------------------------------------------------------------------------
|                          On-Chip Power Summary                            |
-----------------------------------------------------------------------------
|        On-Chip          | Power (mW) |  Used   | Available | Utilization (%) |
-----------------------------------------------------------------------------
| Clocks                  |       0.00 |       0 |    ---    |       ---       |
| Logic                   |       0.00 |     156 |      9312 |               2 |
| Signals                 |       0.00 |     146 |    ---    |       ---       |
| IOs                     |       0.00 |      32 |       232 |              14 |
| Quiescent               |      80.98 |         |           |                 |
| Total                   |      80.98 |         |           |                 |
-----------------------------------------------------------------------------
```

Fig 5: Power Report



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	86	4656	1%
Number of 4 input LUTs	156	9312	1%
Number of bonded IOBs	32	232	13%

Fig 5: Design Summary

CONCLUSION

We propose a Modified rounding based accurate multiplier (MROBA). By modifying the traditional multiplier to get accurate results. Compared to the traditional multiplier, the modified multiplier gives an accurate result for the specified input, and the multiplier can also perform operations that are not in the form of 2n (as performed according to the traditional method), so the exact result for numbers is obtained regardless of 2n. MROBA is designed using Xilinx ISE 14.2 and the results are shown above and other parameters such as area energy delay were calculated using counter-rate and MAC module design is also implemented in Xilinx ISE 14.2.

REFERENCES

[1]. S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, “Energy-efficient accurate multiplication for digital signal processing and classification applications,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[2]. D. R. Kelly, B. J. Phillips, and S. Al-Sarawi, “Accurate signed binary integer multipliers for arithmetic data value speculation,” in Proc. Conf. Design Archit. Signal Image Process., 2009, pp. 97– 10.

[3]. A.B. Kahng and S.Kang,”Accuracy-configurable adder for accurate arithmetic designs,” in Proc 49th Design Autom Conf (DAC), Jun 2012, pp.820- 825.

[4]. S. Deepak, B J kaliath,”Optimized MAC unit Design”.Electron devices and solid state circuit IEEE international conference on 2012.

[5]. P. Siva Nagendra Reddy and M. Saraswathi,” Design and Implementation of FPGA based 64-bit MAC Unit using VEDIC Multiplier and Reversible Logic Gates, Indian Journal of Science and Technology, Vol10(3), DOI:10.17485/ijst/2017/v10i3/ 109413, January 2017.

[6]. Reza Zendegani, Mehdi Kamal, Milad Bahadori, Ali Afzali- Kusha and Massoud Pedram,” RoBA Multiplier: A Rounding-Based Accurate Multiplier for High-Speed yet Energy EnergyEfficient Digital Signal Processing” IEEE transactions on very large scale integration (VLSI) systems syst., pp.1-9, July 2017.