JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

LOW POWER DESIGN OF EDGE DETECTOR USING STATIC SEGMENTEDAPPROXIMATE MULTIPLIERS

1Harish Babu DS, 2Dr. Yogesh G S, 3Dr.Anita R
1Student, 2Professor, HOD, 2Associate Professor
East point college of engineering and technology

Abstract— **In this paper. We recommend a truncation-based approximation multiplier with a compensation circuit made by specific k-map changes to prevent the carry caused by the truncated component. Hardware trimming and output error reduction are completed simultaneously. Using simplified NAND gate circuits, 16x16 truncated multipliers based on sequential approximation multipliers are produced for error correction. We provide a unique segmentation technique for signed SSM. We present a straightforward, hardware-effective multiplier correction method. This design is implemented using Verilog HDL, and it is simulated using ModelSIM 6.4c. To gauge performance, the Xilinx Tool performs the Synthesis Process.**

I. INTRODUCTION

Exact computing units are not always required, especially in applications that can tolerate inaccuracy, including multimedia signal processing and data mining. They are interchangeable with their Seqmented approximate equivalents. study of approximation segmented computing for error-tolerant applications is increasing. The essential elements of these applications are adders and multipliers. They are used in digital signal processing applications and are proposed as transistor-level segmented approximate full adders. Their suggested complete adders are utilized in multipliers to accumulate partial products. Truncation is frequently used in fixed-width multiplier designs to lessen the complexity of multipliers. The quantization mistake caused by the shortened section is then made up for by adding a constant or variable correction term. The accumulation of partial products is the main emphasis of techniques in multipliers, which is essential in terms of power usage. In order to reduce hardware complexity, a array multiplier is built, where the least significant bits of the inputs are truncated while constructing products. The suggested multiplier reduces the number of adder circuits needed to accumulate partial products. There are four different variations of the 8 X 8 Array multiplier, and two designs of 4-2 compressors are described and employed in tree. The suggested compressors' primary flaw is that they produce nonzero output for zero input values, which has a significant impact on the mean relative error (MRE), which will be covered later. The rough design that is suggested in this brief fixes the problem. Better precision results from this. According to a suggested static segment multiplier (SSM), m-bit segments are created from n-bit operands using the operands' leading 1 bit. Then, in place of performing n n multiplication, m m multiplication is carried out. A partial product perforation (PPP) multiplier, where j [0, n-1] and k [1, min (n-j, n-1)], omits k successive partial products starting from the jth position. In [8], a proposed approximate 2 2 multiplier based on altering a Karnaugh map entry is utilized as a building block to create the multipliers 4 4 and 8 8. In [9], an inefficient counter design for the Wallace tree multiplier has been suggested. The presentation of a new rough adder is used for the multiplier's partial product accumulation. Compared to an accurate multiplier, a 16-bit approximate multiplier achieves a 26% power saving. Voltage over-scaling (VOS)-related 8-bit Wallace tree multiplier approximation is discussed. Lowering the supply voltage results in pathways that cannot match the required delays, which causes error.

II. METHODOLOGY

A. Sobel Edge Detection

There are two masks used in Sobel Edge Detection; one mask is used to identify the horizontal edges and the second mask is used to identify the vertical edges. The mask used to compute the vertical edges is comparable to incorporating the gradient in the horizontal direction, and the mask used to discover the horizontal edges is equivalent to having the gradient in the vertical direction.

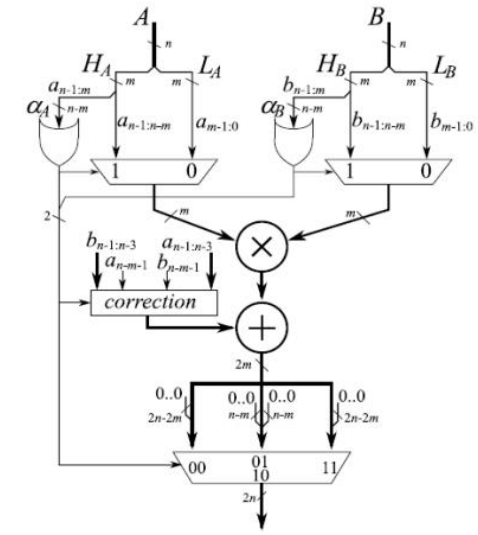
Table.2.1 Sobel detector Mask

-1	-2	-1
0	0	0
1	2	1

1	0	-1
2	0	-2
1	0	-1

B. Block Diagram of Proposed System

Edge detection is a method of image analysis that locates the edges of objects in pictures. To function, it looks for changes in the light. In order to find areas in digital images with significant changes in intensity, discontinuities, or more formally, edges, edge detection employs a variety of mathematical techniques. Edges are a group of curved line segments that are created by the abrupt variations in image intensity. The segmentation approach is used for picture segmentation and data retrieval in a variety of domains, such as image analysis, data analysis, and object recognition. The top three categories of edges are listed below. vertical, outside, and sharp edges.



C. Pre processing

To more swiftly process pictures, we use the Modelsim/MATLAB Co-simulation method. the latter with a straightforward image encoder. The objective is to speed up and improve the efficiency of picture processing. Matlab's tool is being used to convert the image to a Pixel file. The name of this procedure is preprocessing. Preprocessing data is typically the first stage in the deep learning approach to turn raw data into a format that the networks can accept. For example, you can change the input photos to match the proportions of a picture input layer. Preprocessing information can also be utilized to reduce bias-causing artifacts or enhance beneficial qualities. Three Stages of Work are present in this Co simulation.

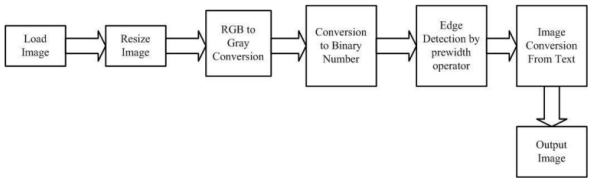
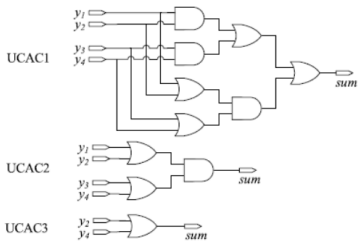


Figure.5. Edge detection based on MATLAB VLSI Co simulation

III. DESIGN OF POWER AND AREA EFFICIENT APPROXIMATE MULTIPLIERS

The majority of signal processing techniques include multiplication as a fundamental operation. Multipliers have a big surface area, a long latency, and use a lot of power. As a result, designing low-power VLSI systems requires careful consideration of low power multiplier design. The multiplier's performance typically determines the effectiveness of a system because it typically consumes the most space and operates at the slowest speed. Consequently, one of the key design considerations is to optimize the multiplier's speed and area. However, area and speed limits frequently conflict, causing larger areas to arise from gains in speed.



The table for the three compressors that are suggested is displayed in below Table.

y_1	y_2	y_3	y_4	UCAC1		UCAC2		UCAC3	
				sum	ED	sum	ED	sum	ED
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	-1	0	-1	1	0
0	0	1	0	0	-1	0	-1	0	-1
0	0	1	1	1	-1	0	-2	1	-1
0	1	0	0	0	-1	0	-1	1	0
0	1	0	1	1	-1	1	-1	1	-1
0	1	1	0	1	-1	1	-1	1	-1
0	1	1	1	1	-2	1	-2	1	-2
1	0	0	0	0	-1	0	-1	0	-1
1	0	0	1	1	-1	1	-1	1	-1
1	0	1	0	1	-1	1	-1	0	-2
1	0	1	1	1	-2	1	-2	1	-2
1	1	0	0	1	-1	0	-2	1	-1
1	1	0	1	1	-2	1	-2	1	-2
1	1	1	0	1	-2	1	-2	1	-2
1	1	1	1	1	-3	1	-3	1	-3

Module Explanation: Reading Image: The Sobel operator is employed to identify edges in the test images. As depicted in Fig. 6, this process is carried out on multiple test images. The picture data is first read as an array with a dimension for the size of the image. To resize the picture array to another array, the array's element count must be computed. In MATLAB, the size that will be used is (256 x 256).

IV. Conclusion

Seqmented approximation multipliers were created. Using the multiplier we have suggested, we will create an efficient array multiplier. The proposed and examined Sequmented approximative compressors for an array multiplier are resented. In the design of a sobel operator, this proposed multiplier is employed. Matlab and the modals program implemented the Sobel operator. Xilinx and modelsim tools.

Table 1. Comparison of proposed work with other works

Proposed MAC Design Summary Settings

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	16	3,840	1%	
Number of 4 input LUTs	206	3,840	5%	
Logic Distribution				
Number of occupied Slices	109	1,920	5%	
Number of Slices containing only related logic	109	109	100%	
Number of Slices containing unrelated logic	0	109	0%	
Total Number of 4 input LUTs	206	3,840	5%	
Number of bonded IOBs	34	97	35%	
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	1,376			
Additional JTAG gate count for IOBs	1,632			

Normal Multiplier MAC Design

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	16	3,840	1%	
Number of 4 input LUTs	246	3,840	6%	
Logic Distribution				
Number of occupied Slices	134	1,920	6%	
Number of Slices containing only related logic	134	134	100%	
Number of Slices containing unrelated logic	0	134	0%	
Total Number of 4 input LUTs	246	3,840	6%	
Number of bonded IOBs	34	97	35%	
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	1,616			
Additional JTAG gate count for IOBs	1,632			

Comparison of Results

	AREA				DELAY		Power
Parameter	Overall Delay	Gate Delay	Path Delay	LUT	Slices	Gates	mW
Dadda Normal	45.844ns	19.138ns	26.706ns	176	97	1056	144
Booth Normal	48.409ns	20.049ns	28.360ns	185	98	1188	141
Proposed	32.628ns	14.730ns	17.898ns	148	78	882	129

References

[1] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and Probabilistic Adders," IEEE Trans. Computers, vol. 63, no. 9, pp. 1760–1771, Sep. 2013.

[2] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," in Proc. Int. Symp. Low Power Electron. Design, Aug. 2011, pp. 409–414.

[3] S. Cheemalavagu, P. Korkmaz, K. V. Palem, B. E. S. Akgul, and L. N. Chakrapani, "A probabilistic CMOS switch and its realization by exploiting noise," presented at the IFIP Int. Conf. Very Large Scale Integ., Perth, Australia, Oct. 2005.

[4] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bioinspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[5] M. J. Schulte and E. E. Swartzlander Jr., "Truncated multiplication with correction constant," in Proc. Workshop VLSI Signal Process. VI, 1993, pp. 388–396.

[5] E. J. King and E. E. Swartzlander Jr., "Data dependent truncated scheme for parallel multiplication," in Proc. 31st Asilomar Conf. Signals, Circuits Syst., 1998, pp. 1178–1182.

[6] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4, pp. 490–501, 2011.

[7] C. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low- power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," IEEE Trans. Circuits Syst., vol. 51, no. 10, pp. 1985–1997, Oct. 2004.