Low Power & Area Multiplier for Deep Learning Applications

Gouri Morankar*

Assistant Professor, Department of Electronics Engineering, Shri
Ramdeobaba College of Engineering & Management, Nagpur, India

ABSTRACT

Multipliers are one of the most important operations in signal and image processing algorithms, artificial intelligence, machine learning algorithms and systems, especially convolutional neural networks in deep learning applications. Therefore it is extremely important to implement hardware intensive multiplier with high speed, regularity in layout, low power and area requirements. These ensure maximum utilization of resources for computational complex algorithms deployed in machine learning applications such as convolutional neural networks. In this paper, design, development and implementation of 16-bit multiplier for hardware intensive application through Baugh Wooley multiplier and Kogge & Stone adder is attempted. Mostly conventional shift-add multipliers are implemented using 2's complement, row & hybrid based adder trees and signal flow optimization techniques, for reduction of partial products through Booth's algorithm. Correspondingly several other techniques for parallel prefix adder are employed for addition in multipliers. However these techniques are less focused on low power and area, regularity in layout and pipelined implementation of multiplier. Proposed 16-bit multiplier employs Baugh Wooley Multiplier and Kogge & Stone adder for addition of partial products that results in pipelined implementation, regularity in layout and achieves higher speed and low power consumption.

KEY WORDS: MULTIPLIER, PARALLEL PREFIX ADDER, LOW POWER, PIPELINED, FPGA IMPLEMENTATION.

INTRODUCTION

Deep learning algorithms through convolutional neural networks (CNN) are computationally demanding with most emphasis on multiplication, division and addition operations. The area and power requirement for these operations is very high and requires considerable time for generating results. Multiplication is one of the most important operations in CNN and plays important role in power consumption and performance of the entire systems. Advancement in integrated circuit technology has enabled researchers and engineers to develop fast,

power and area efficient multipliers that results in low power and area, regularity in layout and pipelined implementation. The common multiplication methods employed are through shifting and adding operations that generates partial and final products. The performance of these multipliers is measured through the number of partial products generated which needs to be added periodically to determine final product.

Modified Booths algorithm through its encoding mechanism reduces partial products by 50% that results in reduced area, delay and power consumption. However it essentially requires complex encoding circuitry which generates partial products. Booths algorithm through 2's complement method was implemented to achieve low power dissipation by increasing the probability of partial products becoming zero. A fixed width array multiplier was designed through left to right algorithm aimed to reduce partial products that resulted in low area and power consumption. Further power reduction in multiplier can be achieved by reducing switching activities. Modified shift-add multiplier namely "bypass zero feed A directly

ARTICLE INFORMATION

*Corresponding Author: morankarg@rknc.edu

Received 19th Oct 2020 Accepted after revision 27th Dec 2020

Print ISSN: 0974-6455 Online ISSN: 2321-4007 CODEN: BBRCBA

Thomson Reuters ISI Web of Science Clarivate Analytics USA and Crossref Indexed Journal

NAAS Journal Score 2020 (4.31)

A Society of Science and Nature Publication,
Bhopal India 2020. All rights reserved.

Online Contents Available at: <http://www.bbrc.in/>

Doi: <http://dx.doi.org/10.21786/bbrc/13.14/101>

(BZ-FAD)” was developed and employed for reducing switching activities and low speed applications. Dynamic error-compensated circuit was designed to reduce the complexity of the encoding circuit and increase the speed through Booths algorithm.

The most important apprehension, the conventional shift-add multiplier is implemented through n-cycles of shifting and adding operation. Mainly the process includes partial product generation and importantly addition of partial products. Various parallel prefix adders were designed and developed by researchers with an intention to reduce area and power of the circuit and achieve higher speed of operation. Fast and area efficient adders are developed with different adder structures based on carry lookahead adder (CLA) and Ling equation and its modified versions. Modified CLA equation was reorganized by Kogge and Stone to increase the speed of the adder which can also achieve regularity in layout and employed for summation of partial products in add shift multiplier. Baugh Wooley multiplication algorithm [6] was designed and developed using regular 2's complement multipliers which achieves high speed and regularity in layout. It efficiently handles sign bit for computation of partial products with the use of full adder and ripple carry adder.

In this paper, design, development and implementation of 16-bit multiplier for hardware intensive application through Baugh Wooley multiplier and Kogge & Stone adder is attempted. Mostly conventional shift-add multipliers are implemented using 2's complement, row & hybrid based adder trees and signal flow optimization techniques, for reduction of partial products through Booth's algorithm. Correspondingly several other techniques for parallel prefix adder are employed for addition in multipliers. However these techniques are less focused on low power and area, regularity in layout and pipelined implementation of multipliers. Proposed 16-bit multiplier employs Baugh Wooley Multiplier and Kogge & Stone adder for addition of partial products that result in pipelined implementation, regularity in layout and achieves higher speed and low power consumption. The paper is organized as section 1 introduces to design and development of add shift multipliers and parallel prefix adders. Section 2 discusses the design of Baugh Wooley multiplier and parallel prefix adder using Kogge & Stone adder that results in high speed, low power and regularity in layout. Results are discussed in section 3 and finally concluded in section 4.

2. Modified Baugh Wooley Multitplier using Kogge & Stone Adder

Let a and b be the two n-bit signed numbers which can be represented as,

$$a = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} 2^i a_i \quad (1)$$

$$b = -b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} 2^j b_j \quad (2)$$

The result of multiplication of a and b is represented as

$$\begin{aligned} p &= a \times b \\ &= \left(-a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} 2^i a_i \right) \times \left(-b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} 2^j b_j \right) \\ &= a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2} 2^i a_i \sum_{j=0}^{n-2} 2^j b_j - \\ &\quad a_{n-1}2^{n-1} \sum_{j=0}^{n-2} 2^j b_j - b_{n-1}2^{n-1} \sum_{i=0}^{n-2} 2^i a_i \end{aligned} \quad (3)$$

The last two terms in equation (3) are n-1 bits each that are extended from position $2n-1$ to $2n-3$. Rather than subtracting the last two terms, 2's complement was obtained through last two terms and final product was obtained using addition of partial products. Let p be one of the last two terms, it can be represented in equation (4) with zero padding.

$$p = -0 \times 2^{2n-1} + 0 \times 2^{2n-2} + 2^{n-1} \sum_{j=0}^{n-2} 2^j z_j + 0 \times \sum_{r=0}^{n-2} 2^r \quad (4)$$

2's complement of p are obtained, the new bit value for -p is depicted in table 1.

Table 1. Bit values for -p

Bit position	Bit Values
2n-1	1
2n-2	1
2n-3	Z_{n-2}
2n-4	Z_{n-3}
2n-5	Z_{n-4}
...	...
n	Z_1
n-1	Z_0+1
...	...
1	0
0	0

Table 2. Bit patterns

Bit position	2n-1	2n-2	n	n-1
+	1	1		1
	1	1		1
Carry in	1	0 / 1	1	0 / 1
Sum		0 / 1		0 / 1

Let z_1 and z_2 be last two terms in equation (3) then addition of $-z_1 + (-z_2)$ results in following various bit patterns depicted in table 2. Hence the product p in equation (3) can be given as.

$$\begin{aligned} p &= a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} 2^{i+j} a_i b_j + 2^{n-1} \sum_{j=0}^{n-2} 2^j \overline{b_j a_{n-1}} \\ &\quad + 2^{n-1} \sum_{i=0}^{n-2} 2^i \overline{a_i b_{n-1}} - 2^{2n-1} + 2^n \end{aligned} \quad (5)$$

Fig. 1 depicts the structure of 4-bit Baugh Wooley multiplier with ripple carry adder which can be potentially replace with Kogge & Stone adder and fig. 2 depicts the internal structure of Baugh Wooley Cells which contents full adder and does not need replacement as carry is propagated to next stage. The design of Kogge & Stone adder can be comprehended using three steps process which includes pre-processing, carry look ahead computation and post processing depicted in fig. 3. The pre-processing steps involves computation of propagate (p) and generate (g) bits using equation (6) and (7) respectively.

$$p_i = a_i \text{ xor } b_i \quad (6)$$

$$g_i = a_i \text{ and } b_i \quad (7)$$

Carry look ahead computation involves calculation of carry's corresponding to each bit. It uses propagate (p) and generate (g) bits as partial results which are fed to the carry equations for fast computation as depicted in below equations.

$$P = p_i \text{ or } (g_i \text{ and } p_{i-1}) \quad (8)$$

$$G = g_i \text{ and } g_{i-1} \quad (9)$$

Finally the post processing calculates final sum of the addition operation using equation (10).

$$s_i = p_i \text{ xor } c_{i-1} \quad (10)$$

Figure 1: Structure for 4-bit Baugh Wooley Multiplier with ripple carry adder which can be potentially replace with Kogge & Stone adder

3.Experimental Results: The 16-bit synchronous pipelined Baugh Wooley multiplier using Kogge Stone adder was coded in VHDL and implemented in virtex 5 FPGA (xc5v1x20t-2ff323) device. The output of the 16-bit multiplier was verified using Xilinx ISE web pack 13.1 simulation and synthesis tool. Table 3 summaries

of resource utilization, delay, operating frequency and power consumption.

Figure 2: Internal Structure of Baugh Wooley Cells which contents full adder and does not need replacement as carry is propagated to next stage

Figure 3: Internal structure of 8-bit Kogge & Stone adder

Table 3. Experimental results obtained after simulation, synthesis and implementation on virtex 5 FPGA (xc5v1x20t-2ff323) devices

FPGA Resources/ Parameters	BWM using full adder 16 x 16	BWM using KS adder 16 x 16	No pipelined BWM 8 x 8
Number of Slices	515	490	55
Number of LUTs	503	478	114
Number of FFs	526	526	-
Delay	2.844 ns	2.67 ns	4.172 ns
Maximum Frequency	351.58 MHz	374.53 MHz	239.69 MHz
Clock Frequency	1 GHz	1 GHz	-
Total Power	1.758 W	1.669 W	0.224 W
Dynamic Power	1.421 W	1.349 W	-
Leakage Power	0.337 W	0.32 W	-

the results obtained after simulation, synthesis and implementation. Table 1 also demonstrates comparison with fully synchronous Baugh Wooley Multiplier (BWM) using full adder and Kogge & Stone (KS) adder in terms

The results clearly depicts maximum operating frequency of 351 MHz using ripple carry adder whereas maximum operating frequency of 375 MHz using Kogge & Stone adder with marginally improvement in resource utilization. Implementation results are also compared with no pipelined balanced BWM which clearly demonstrates superior performance of BWM using KS adder in terms of speed. Pipelined implementation of 16-bit synchronous pipelined Baugh Wooley multiplier using Kogge Stone adder in 18 steps dissipates 1.669 W of total power with improvement in 5% compared with conventional implementation.

CONCLUSION

In this paper, design, development and implementation of 16-bit multiplier for hardware intensive application through Baugh Wooley multiplier and Kogge & Stone adder was attempted. Mostly conventional shift-add multipliers are implemented using 2's complement, row & hybrid based adder trees and signal flow optimization techniques, for reduction of partial products through Booth's algorithm. Correspondingly several other techniques for parallel prefix adders are employed for addition in multipliers. However these techniques are less focused on low power and area, regularity in layout and pipelined implementation of multipliers. Proposed 16-bit multiplier employs Baugh Wooley Multiplier and Kogge & Stone adder for addition of partial products that result in regularity in layout and pipelined implementation and achieves higher speed and low power consumption. The multiplier is modelled using VHDL Xilinx ISE web pack 13.1 and implemented using vertex 5 FPGA (xc5vlx20t-2ff323) device. Comparison with conventional implementation results improvement in area, power consumption and pipelined implementation through regularity of layout. It clearly demonstrates that signed Baugh Wooley multiplier with Kogge Stone adder results in better implementation and may be employed in deep learning algorithms using convolutional neural network. Further improvement in optimization in signal flow graph may further improve delay and power consumption.

REFERENCES

- A. Badawi, A. Alqarni, A. Aljuffri, M. S. BenSaleh, A. M. Obeid and S. M. Qasim, "FPGA realization and performance evaluation of fixed-width modified Baugh-Wooley multiplier," 2015 Third IEEE International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE), Beirut, 2015, pp. 155-158, doi: 10.1109/TAECE.2015.7113618.
- B. Shao and P. Li, "Array-based approximate arithmetic computing: A general model and applications to multiplier and squarer design," in IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 62, pp. 1081-1090, April 2015, doi: 10.1109/TCSI.2015.2388839.
- D. Giorgos and D. Nikolos, "High-speed parallel prefix VLSI ling adders," in IEEE Transactions on Computers, Vol. 54, No. 2, Feb 2005, pp. 225-231, doi: 10.1109/TC.2005.26.
- D. Kalaiyarasi and M. Saraswathi, "Design of an efficient high speed radix-4 Booth multiplier for both signed and unsigned numbers," 2018 4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, India, 2018, pp. 1-6, doi: 10.1109/AEEICB.2018.8480959.
- F. U. D. Farrukh, T. Xie, C. Zhang and Z. Wang, "Optimization for Efficient Hardware Implementation of CNN on FPGA," 2018 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA), Beijing, China, 2018, pp. 88-89, doi: 10.1109/CICTA.2018.8706067.
- Gouri Wazurkar & S. L. Badjate, "Globally Asynchronous Locally Synchronous (GALS) Pipelined Signed Multiplier", 2016 IEEE International Conference on Computing, Analytics & Security Trends, College of Engineering, Pune, 2016, pp. 19 - 21, doi: 10.1109/CAST.2016.7914999.
- J. S. Wang, C. N. Kuo, and T. H. Yang, "Low power fixed-width array multipliers," in 2004 Proceedings of International Symposium on Low Power Electronics and Design, 2004, pp. 307-312, doi: 10.1109/LPE.2004.241092.
- M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram, "BZ-FAD: A low-power low-area multiplier based on shift-and-add architecture," in IEEE Transactions on Very Large Scale Integration(VLSI) Systems, Vol. 17, No. 2, Feb 2009, pp. 302-306, doi: 10.1109/TVLSI.2008.2004544.
- N. Poornima and V. S. Kanchana Bhaaskaran, "Area efficient hybrid parallel prefix adders," in Procedia Materials Science, Vol. 10, 2015, pp. 371-380, doi: 10.1016/j.mspro.2015.06.069.
- N. Y. Shen and O. C. Chen, "Low - power multipliers by minimizing switching activities of partial products," 2002 IEEE International Symposium on Circuits and Systems, Vol. 4, 2002, pp. 93-96, doi: 10.1109/ISCAS.2002.1010397.
- P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," IEEE Transactions on Computers, Vol. 22, No. 8, 1973, pp. 786-793, doi: 10.1109/TC.1973.5009159.
- W. Q. He, Y. H. Chen, and S. J. Jou, "Dynamic error compensated fixed-width Booth multiplier based on conditional-probability of input series," 2016 Circuits, Systems, and Signal Processing, Vol. 35, No. 8, pp. 2972-2991, doi.org/10.1007/s00034-015-0186-2
- Xilinx, "Virtex-5 Family Overview," [online]; <http://www.xilinx.com>, v5.1, 2015.