Design of High Speed Mac Unit

[1]Harish Babu N, [2]Rajeev Pankaj N

[1]PG Student, [2]Assistant professor
Schools of Electronics Engineering, VIT University, Vellore -632014, TamilNadu, India.
[1]harishharsha72@gmail.com, [2]rajeevpankaj@vit.ac.in

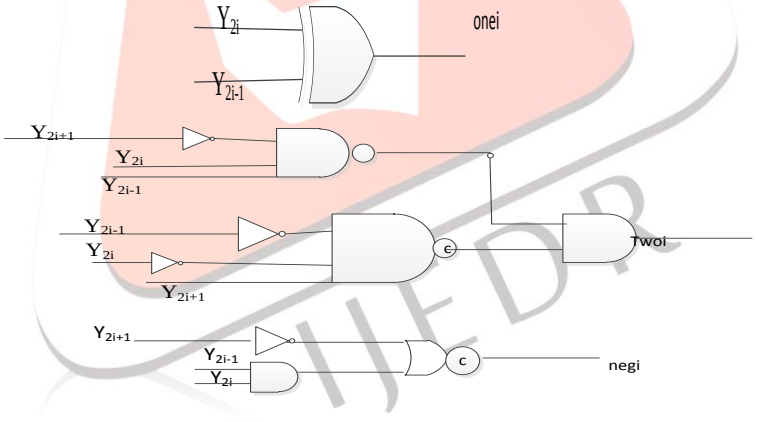
***Abstract*— Radix-4 technique is most employed technique to reduce the power consumption and delay when compared to all other techniques in which we use the booth recoding table for obtaining partial products thus it can be used in a MAC unit. By using carry select adder the partial products are added. By using this RTL coding for 8×8 and 16×16 32×32 using multiplication with modified booth technique and the MAC units for 8×8 16×16, 32×32 the Simulation is performed in ModelSim. Pipelining is implemented with two cycles to improve the speed of the MAC Unit. The synthesis carried in (Cadence) rc and area, delay, power report are obtained.**

***Index Terms*— Radix-4 multiplication, neg3 addition, 2 cycle pipelining.**

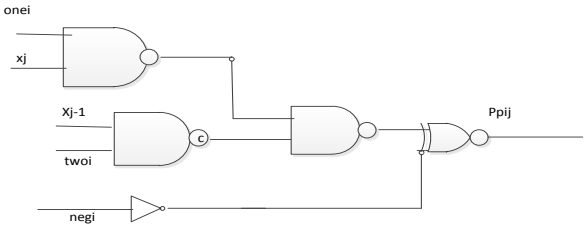
I. INTRODUCTION

Multiplier is the important block in almost all the arithmetic logic units. These multipliers are mostly used in the fields of the Digital Signal Processing (DSP), Fast Fourier Transform, convolution, filtering and microprocessor applications. Since multiplier is the main component and hence a high speed and area efficient multiplier is needed to achieve this one of the finest technique is by using Radix -4 encoding.

Radix-4 MBE is mostly used because it reduces total number of partial product rows to half while maintaining the generation of each partial product row fast. If 'n' is the total number of bits in multiplier, radix-4 MBE technique reduces these partial product rows by "$\frac{n}{2}$+1".By using the proposed technique we are making that "$\frac{n}{2}$+1" partial products to "$\frac{n}{2}$" By adding neg3 at the top of the first partial product row.



1(a)



1(b)

Figure 1. Gate-level diagram for partial product generation using MBE
(a) MBE signals generation.(b) Partial product generation

The MBE signal generation unit is called as encoding technique, partial product generation technique is called as a decoding scheme after obtaining the partial products we should add them to add it we are using the carry select adder module[2]. After summing of all the partial products we can get the desired output by adding the vectors in which we can use the Modified carry select adder to increase the speed of operation and to reduce the power consumption [1]. Modified booth algorithm consumes less power when compared to other type of multiplication algorithms because of using Modified sqrt carry select adder circuit by

replacing the ripple carry adder circuit by BEC Unit which inbuilt reduce the power consumption as an incremental unit .By applying pipelining the delay gets further reduced.

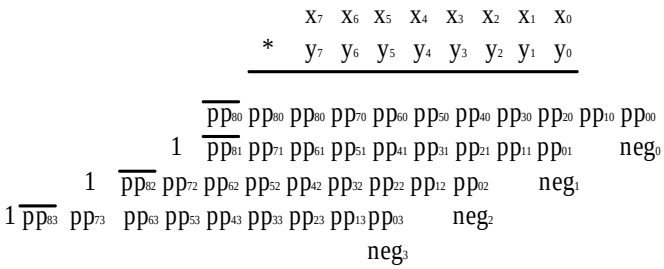


Figure 2. Sign extension prevention measure on the partial product array of a 8×8 radix-4 MBE multiplier[2].

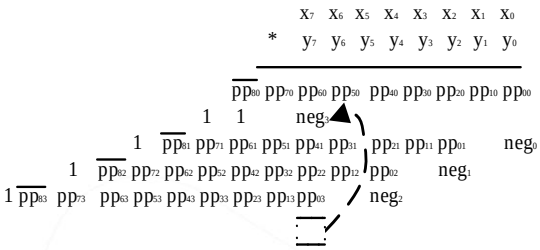


Figure 3. Idea for reducing the partial product array[2].

The main idea to reduce the power consumption is by using reduction of partial product array. Thus speed and power can be improved by using these neg3 bit to add at the top of the first partial product row[3].

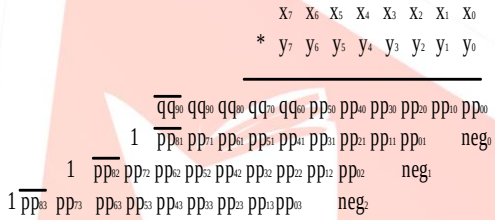


Figure 4. Resultant rows after incorporating the idea

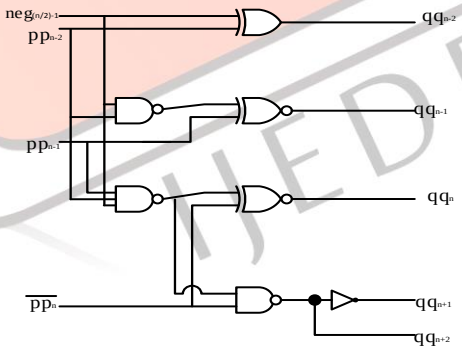


Figure 5. The method for adding the last Neg bit in the first row[2]

II. ADDING THE PARTIAL PRODUCTS USING CARRY SELECT ADDER

Normal carry select adder

In this carry select adder circuit we are using the two ripple carry adder blocks parallel in which it is having more power consumption because of more hardware circuitry. Here in this carry select adder circuit we are replacing the normal ripple adder block with Binary to excess one converter. In which the binary to excess one module consists of adding one to ripple result which reduces the no of gates so the power consumption decreases, and speed increases to high level.

By using this in partial products addition the results are better optimized with respect to power and delay when compared with other type of adders. The reduced gates can be shown in the form of equations as

eo=~bo
e1=bo^b1
e2=b2^ (bo&b1)
e3=b3^ (bo&b1&b2)
cx=(b0&b1&b2&b3)

The above expressions we get it from K-Map[1]

Modified sqrt carry select adder

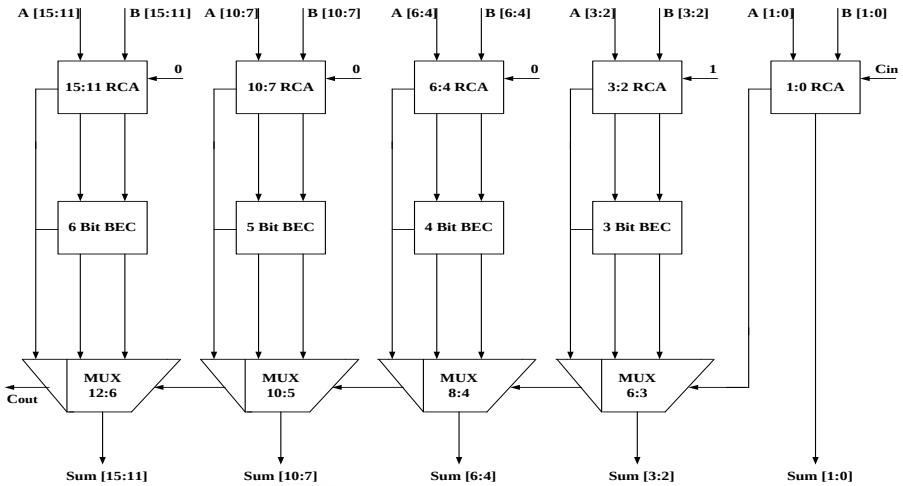


Figure 6. Modified sqrt carry select adder[1]

III. BASIC MAC UNIT

In this MAC Unit we are generating the products and adding them using a accumulator register to store the values by using clock. Example of MAC Expression AB+CD+EF which can be obtained from Fig 7 with pipelined architecture. Whenever pipelining is employed the critical path gets reduced and throughput of the architecture gets improved which inbuilt increase the speed. Pipelining is implemented in two clock cycles, Whenever pipelining is implemented using an extra register the speed gets increased but there is a change in power constraints also[4]. By increasing the speed of the unit there is little bit increase in power constraints.

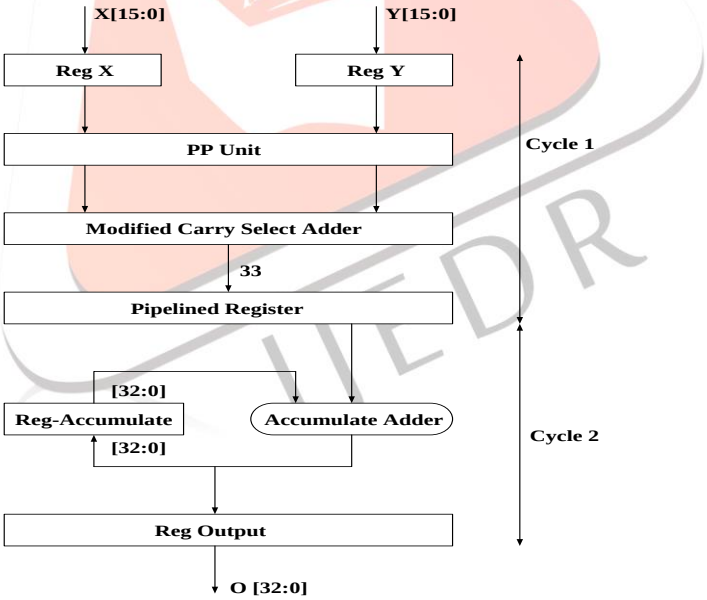


Figure 7. MAC Architecture

The 2-cycle MAC Unit is having a critical path which goes through the PP unit and the final adder a pipeline register is placed before accumulator by copying into temporary register which has no impact on functionality. By using the pipelining, our MAC unit gives the correct output in each clock cycle, and no extra clock pulses to be added at the end of the loop. By incorporating several guarding bits, making longer devices feasible without making any overflow considerations. So the use of guarding bits in a technique where the accumulated is send back to the PP's input, which would most certainly have a negative impact on hardware circuitry which makes architecture complex to solve the issues regarding guard bits. Thus pipelining provides increase in fastness of the device by 28% with a slight increase of power consumption, whenever we attempt for 3 cycles then power consumption increase to maximum level in order to minimize it the two cycle configuration is better employed in the MAC architecture.

IV. RESULTS AND ANALYSIS

The neg3 method addition gives better results when compared to other multiplication techniques which give less delay consumption when compared with modified booth algorithm, but this delay is more than Vedic, array multiplication techniques but better than baugh wooley multiplication [6]. So in order to improve the speed pipelining is implemented so that MAC Unit operates with high throughput but little bit increase with power consumption, because the increase in power consumption doesn't affect the performance because modified booth multiplication takes minimum power consumption next to baugh wooley multiplication. So by implementing pipelining the increase in power consumption is very less when compared to all other MAC Units [7].

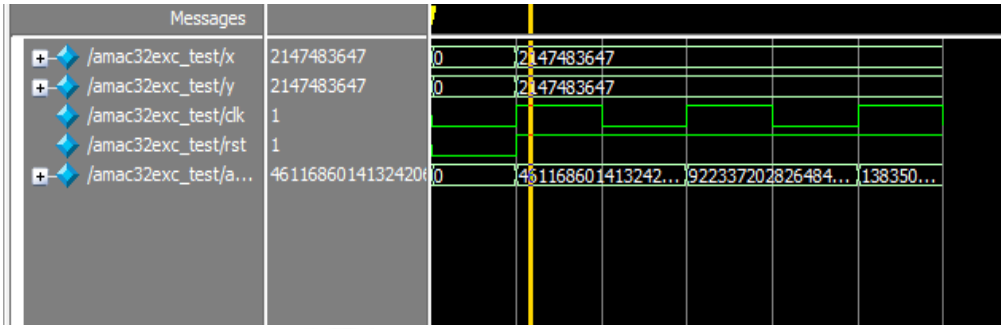


Figure 8. MAC without pipelining

Multiplication type	Delay(ns)	Power(nW)
Modified Booth 8 X8	3.4	59603.93
Negi method 8 X8	3.04	59923.47
Modified Booth 16X16	6.85	96555.376
Negi method 16X16	6.37	107752.864
Modified Booth 32X32	13.74	154572.45
Negi method 32X32	12.64	155572.81

a. Comparison of Power, Delay in multiplication

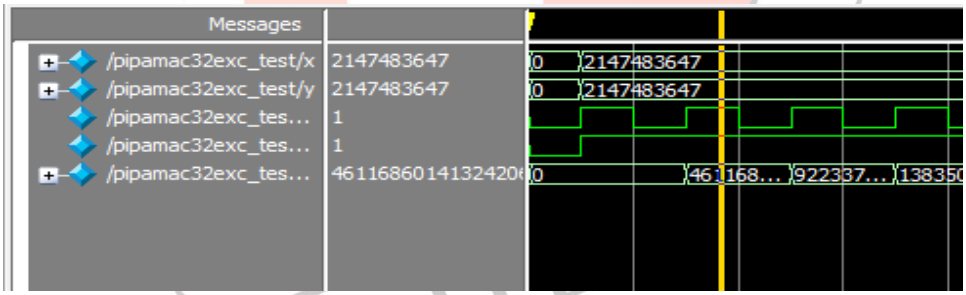


Figure 9.MAC with pipelining

MAC Unit	Delay ns	Area (No of cells)	Total Power (nW)
without pipelining 8X8	3.846	463	100754.93
With pipelining 8X8	2.724	496	126787.14
without pipelining 16X16	8.245	1381	227687.57
with pipelining 16 X16	5.934	1674	264625.41
without pipelining 32X32	17.09	2904	585123.87
with pipelining 32X32	12.68	3211	665923.45

b. Comparison of Power, Delay in and MAC Unit

V. CONCLUSION

The Pipelined MAC Unit gives lesser delay when compared to all other MAC Units and reasonable power consumption because the negi method of Modified Booth multiplier gives less power consumption when compared to other multiplication algorithms since Modified carry select adder is used for adding the partial products. so by applying the pipelining the delay also gets reduced and shows better performance.

VI. ACKNOWLEDGMENT

We thank the School of Electronics Engineering, VIT University, Vellore for providing the tool Cadence environment and giving us valuable suggestions.

REFERENCES

[1] R.Priya, J.Senthil Kumar, "Implementation and Comparison of Effective Area Efficient Architectures for CSLA" 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), 2013

[2] Fabrizio Lamberti, Nikos Andrikos, Elisardo and Paolo Montuschi, "Reducing the Computation Time in (Short Bit-Width) Two's Complement Multipliers", IEEE Transactions on Computers, Vol. 60, No. 2, February 2011.

[3] F. Lamberti, N. Andrikos, E. Antelo, and P. Montuschi, "Speeding-Up Booth Encoded Multipliers by Reducing the Size of Partial Product Array," internal report, http://arith.polito.it/ir_mbe.pdf, pp. 1-14, 2009.

[4] Tung Thanh Hoang, Magnus Sj¨alander, and Per Larsson-Edefors "High-Speed, Energy-Efficient 2-Cycle Multiply Accumulate Architecture", pp 978-1, IEEE2009.

[5] J.-Y. Kang and J.-L. Gaudiot, "A Fast and Well-Structured Multiplier," Proc. Euromicro Symp. Digital System Design, pp. 508-515, Sept. 2004.

[6] Magnus Själander and Per Larsson-Edefors, "High-Speed and Low-Power Multipliers Using the Baugh-Wooley Algorithm and HPM Reduction Tree ", 2008 IEEE

[7] Debaprasad Das and Hafizur Rahaman, "A Novel Signed Array Multiplier" 2010 International Conference on Advances in Computer Engineering , 2010 IEEE.

[8] Dr. K.S. Gurumurthy , M.S Prahalad, " Fast and Power Efficient 16×16 Array of Array Multiplier using Vedic Multiplication"

[9] Morris Mano, "Computer System Architecture", 3rd edition, Prientice-Hall, New Jersey, USA, 1993, pp. 346-348