International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

Utilization of Memory by using APC – OMS Approach

Kuruba.Mamatha & K.Thriveni
[1] *M.Tech student, Department of ECE, BIT Institute of Technology, Hindupur.*
[2] *Assistant Professor, Department of ECE, BIT Institute of Technology, Hindupur.*

ABSTRACT
Recent trends in Electronics had advanced to three mutual conflicting demands they are high Low power consumption, computational-bandwidth and reconfigure ability. It is a challenging task to figure out these conflicting demands. The optimization of a filter has a great impact in the hardware design. This optimization can be achieved by using various techniques in multipliers and adders. The efficient way of optimizing multiplier is carried out by implementing a Lookup table Multipliers that performs the fast computations with reduced area and power. Generally, by using Anti symmetric product coding (APC) and Odd multiple storage (OMS) techniques we reduce the conventional multiplier to one-fourth of its size. This reduced multiplier can be further reduced by implementing a Generalized version of a Anti symmetric product coding (GAPC) in a filter. Therefore, this modified combination of GAPC and OMS reduces the size of an LUT multiplier.

Keywords
Anti-Symmetric Product Coding (APC), Odd Multiple storage (OMS), Generalized Anti-symmetric product coding (GAPC), Lookup table (LUT).

INTRODUCTION
A filter is said to be an efficient filter if it performs Fast computations with reduced area, power and increase in performance. In the world of Digital Processing, the processing of digital information with minimum noise plays a major role for the use of filters. The computations in digital systems increase with decreasing size of the hardware. The conventional way of multiplication is nothing but the repeated additions that consumes more power and hardware size. Using of optimized lookup tables techniques in the filter could be of great benefit in future digital world.

The different techniques used are Anti-Symmetric product coding scheme (APC), Odd-Multiple Storage scheme (OMS), Combined approach, and Generalized Anti-symmetric product coding (GAPC).Combined approach is nothing but the combination of both Anti symmetric product coding (APC) and a Modified Odd multiple storage (OMS) techniques. Instead of using 32 different memory locations we use only 8 different locations and retrieve all the remaining values. Therefore, reduces the size to one-fourth of the conventional multiplier.

COMBINED APPROACH
In this, a 5-bit is given as an input to the anti-symmetric product coding block. The computation of the different product word values (i.e., input word [X] of length L=5 multiplied by fixed coefficient value A) is then preceded to the odd multiple storage, which is finally either subtracted or added with the constant coefficient 16A. This entire process facilitates in reducing the size of an LUT.

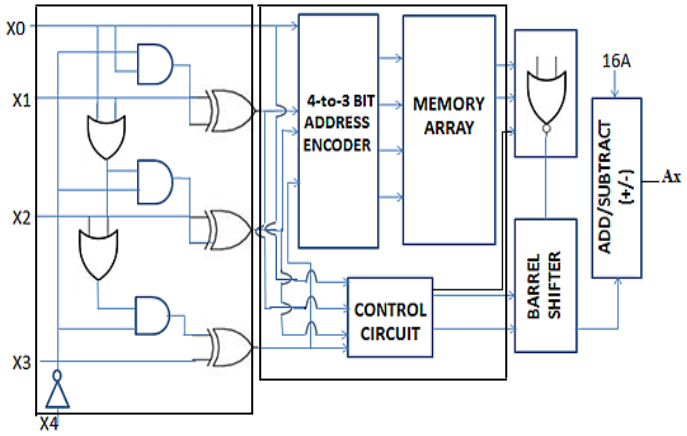

Fig 1: Combination of APC& Modified OMS Approach

ANTI-SYMMETRIC PRODUCT CODING
The Anti symmetric product coding scheme behaves the Negative mirror symmetric property of the bits. Initially, an input of 5bit length is given. Generally, for n bit input, there will be 2^n number of combinations for a conventional LUT multiplier that uses the total 32 different Memory locations for storing of Values.

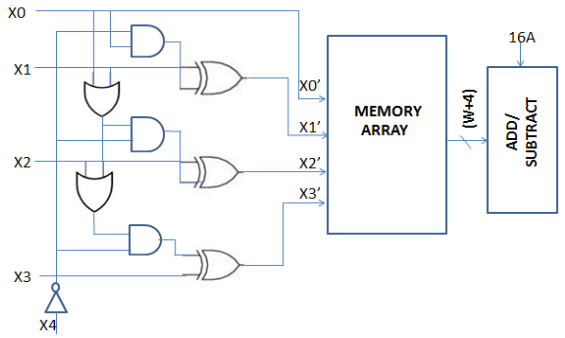

Fig2. Antisymmetric product scheme

In APC scheme, it provides a Negative symmetric property for which the input is replicated as two's complement. Therefore, instead of using 32 different memory locations only 16 different memory locations are used.

Here, we can consider a 16A as a constant-coefficient through which we can either add or subtract the input to generate the remaining half inputs using the addition or subtraction operation. From the table below, we observe the input word X on the first column of each row is the two's complement of that on the third column.. In addition, the sum of the product values corresponding to these two input values on the same row is 32A.

Instead, consider the product values on the second and fourth column of a row be u and v respectively:

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

We can write it as [1].

$$u = \left[\left(\frac{u+v}{2}\right) - \left(\frac{v-u}{2}\right)\right] \quad (1a) \quad \text{and}$$

$$v = \left[\left(\frac{u+v}{2}\right) + \left(\frac{v-u}{2}\right)\right]. \quad (1b)$$

For (u+v) =32A, where 'A' is a fixed coefficient. Therefore, we have

$$u = \left[16A - \left(\frac{v-u}{2}\right)\right] \quad (2a)$$

$$v = \left[16A - \left(\frac{u-v}{2}\right)\right] \quad (2b)$$

The 4-bit LUT addresses and corresponding APC words are listed in (table1)

Table1: 4bit address and its APC

Input	Product value 'X'	Input	Product value 'X'	Address	Apc Word
00001	A	11111	31A	1111	15A
00010	2A	11110	30A	1110	14A
00011	3A	11101	29A	1101	13A
00100	4A	11100	28A	1100	12A
00101	5A	11011	27A	1011	11A
00110	6A	11010	26A	1010	10A
00111	7A	11001	25A	1001	9A
01000	8A	11000	24A	1000	8A
01001	9A	10111	23A	0111	7A
01010	10A	10110	22A	0110	6A
01011	11A	10101	21A	0101	5A
01100	12A	10100	20A	0100	4A
01101	13A	10011	19A	0011	3A
01110	14A	10010	18A	0010	2A
01111	15A	10001	17A	0001	A
10000	16A	10000	16A	0000	0

Consider, for a given 5-bit addresses the corresponding code words to be stored are reduced to half. This is derived from

the anti-symmetric behavior of products, representing the negative mirror symmetry [1]. The representation of the address bits is, such that X(x0, x1, x2, x3, x4)

If x4=1; then $X' = X_L$ then

if x4=0; then $X' = XL'$

Where X_L is the four less significant bits of X, and X_L' is the two's complement of XL. The product word can be denoted as:

PW= 16A+ (sign value) * (APC word) (3)

Where,

Sign Value = +1 for x4=1 and,
Sign Value = -1 for x4 = 0.

The product value for X= (10000) corresponds to the anti-symmetric product code value "zero", which could be derived by resetting the LUT output, instead of storing it in LUT.

ODD MULTIPLE STORAGE

The Look-Up Table based multiplier adapts the Odd Multiples Storage (OMS) scheme where the memory locations are used to store the Odd multiple values.

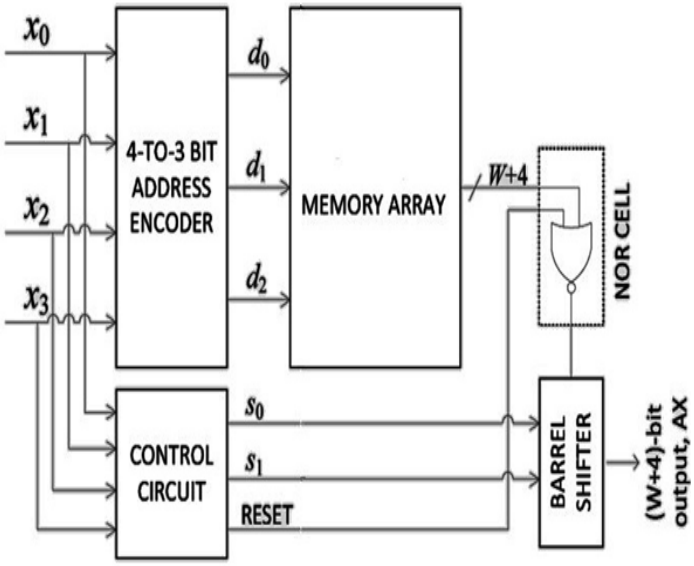

Fig3. Odd Multiple Storage block diagram

Here x0, x1, x2 and x3 are the 4-bit input streams for the multiplier while the multiplier of 4-bit is 'A', which is a fixed coefficient. The 4-bit input stream, is given as input to the 4-to-3 address encoder. The encoder converts the given data into a 3-bit address. In order to store the bits a lookup table or a simple 3to8 decoder can be used for storing in different memory allocations. Nor gate is used to Set or RESET the signal.

The control unit performs the switching activities that controls both barrel shifter and NOR cell using s0, s1 lines. From the below tabular column we observe that only odd multiple values are stored in the address generated instead of storing all the product words.

Table2: Odd multiple storage

Address word	Product word	Address word	Product word	Address word	Product word
0000	0	1000	8A	0001	A
0001	A	1001	9A	0011	3A
0010	2A	1010	10A	0101	5A
0011	3A	1011	11A	0111	7A
0100	4A	1100	12A	1001	9A
0101	5A	1101	13A	1011	11A
0110	6A	1110	14A	1101	13A
0111	7A	1111	15A	1111	15A

Thus, the combined approach reduces the size of a Lookup table multiplier to one-fourth of its original (Conventional) size. The optimization of LUT is further preceded by generalizing the different techniques. In this, the Anti

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

symmetric product coding is further generalized to 8-different memory locations where instead of storing in 16 different locations only 8 memory locations are used.
The generalization is possible with two control signals which are preceded at the last two most significant bits. The entire system is controlled with x3 and x4 control signals.

3. GENERALIZED ANTI-SYMMETRIC PRODUCT CODING

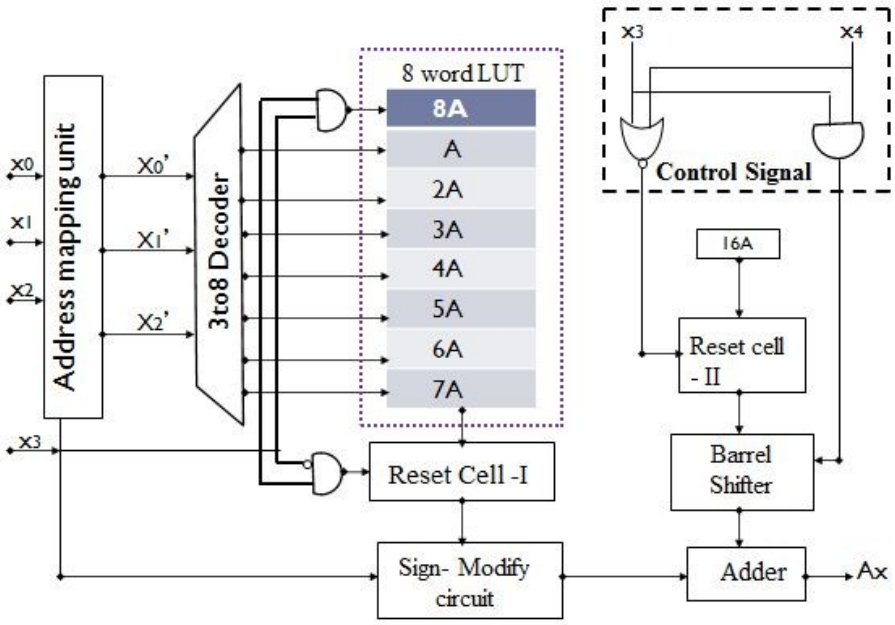

Fig4. Generalized Anti-symmetric product coding scheme

The product words for different values of X for Length L=5 can be estimated using the control signals as shown in the block diagram. Here, we store only 8 values and retrieve the remaining values by considering two constant values as 16A and 32A.

The key control signals are the x3 and x4, it performs certain conditions such as:
For x3 =0, the GAPC encoded word (x3' x2' x1' x0') = (x3 x2 x1 x0) and,
For x3 = 1, the GAPC encoded word (x3' x2' x1' x0') is equal to its 2's complement of (x3 x2 x1 x0) again,
For x3 = 0 and x4 = 0: AX = AX'.
For x3 = 1 and x4 = 0: AX = 16A - AX'.
For x3 = 0 and x4 = 1: AX = 16A + AX'.
For x3 = 1 and x4 = 1: AX = 32A - AX'.
The multiplication of any word X of size L, with the fixed coefficient A, for possible combinations of **2^L** values of C=A.X, only those words may only be stored in the LUT as follows:

- The barrel-shifter producing a maximum of (L-1) left-shifts is used to derive (or) retrieve the remaining even multiples of A.
- The L-bit input word that maps to (L-1) bit address of the lookup table by the Address encoder, control-bits for barrel-shifter is derived by the control-circuit itself.

Table3: Generalized Anti-symmetric product coding

Input, X	Encoded word, X'	GAPC values	Normal values	Input, X	Encoded word, X'	GAPC values	Normal values
00000	0000	0	0	10000	0000	0	16
00001	0001	A	1	10001	0001	A	17
00010	0010	2A	2	10010	0010	2A	18
00011	0011	3A	3	10011	0011	3A	19
00100	0100	4A	4	10100	0100	4A	20
00101	0101	5A	5	10101	0101	5A	21
00110	0110	6A	6	10110	0110	6A	22
00111	0111	7A	7	10111	0111	7A	23
01000	1000	8A	8	11000	1000	8A	24
01001	0111	7A	9	11001	0111	7A	25
01010	0110	6A	10	11010	0110	6A	26
01011	0101	5A	11	11011	0101	5A	27
01100	0100	4A	12	11100	0100	4A	28
01101	0011	3A	13	11101	0011	3A	29
01110	0010	2A	14	11110	0010	2A	30
01111	0001	1A	15	11111	0001	1A	31

These different LUT multiplier techniques can be further implemented in Digital signal processing applications which

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

would be an effective way of satisfying the conflicting demands they are consuming less power, increasing the performance (Computation) and the size of the hardware. It is then implemented in a four tap FIR filter. The multiplier section of the filter circuits are implemented with the combined approach multipliers.

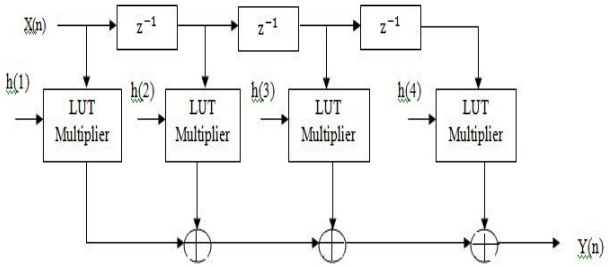

Fig5: FIR Filter Design

The simulation works are carried out in Model Sim Version 6.3 simulator and the functionality of the designs are verified from the waveforms generated.

The simulated waveforms of a lookup table Multiplier are:

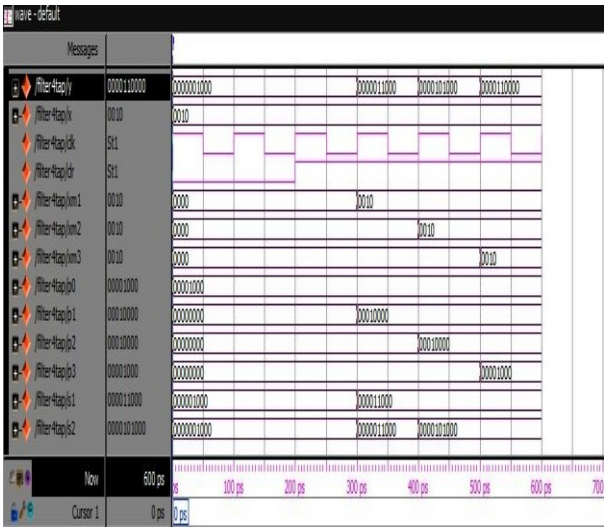

Fig6: APC – OMS LUT Multiplier

CONCLUSION

Implementing these techniques for different lengths is also possible considering 0A, 32A, 64A as the constant, depending on their lengths. Therefore, implementing these techniques in LUT based multiplier will be an area efficient alternative for other memory based computation methods and requires comparatively less time for processing than the conventional multipliers and therefore can offer high speed operations and is more suitable for High speed low level DSP applications that deal with constant coefficient digital filters.

REFERENCES

[1] "LUT-based Circuits for Future wireless systems" by P.K. Meher, 2010, 978-1-4244-7773-9.
[2] P.K. Meher, "LUT Optimization for Memory-Based Computation" in Circuits and SystemII, April. 2010, pp. 285–289.
[2] H.-C. Chen, J.-I. Guo, T.-S. Chang, and C.-W. Jen, "A memory efficient realization of cyclic convolution and its application to discrete cosine transform"
[3] John G. Proakis and Dimitris G. Manolakis, Digital Signal Processing Principles, Algorithms and Applications – third edition, Prentice-Hall of India Pvt. Ltd., New Delhi, 2005.
[4] P.K. Meher, "Memory-based hardware for resource constrained digital signal processing systems," in Proc. 6th Int. Conf. ICICS, Dec. 2007
[5] P. K. Meher, "New approach to LUT implementation and accumulation for memory-based multiplication," in Proc. IEEE ISCAS, May 2009, pp. 453–456.
[6] Pramod Kumar Meher, "New Look-up-Table Optimizations for Memory-Based Multiplication," in Proc. 12th Int. Conf. ISIC, Dec. 2009.
[7] Keshab K. Parhi- University of Minnesota, VLSI Digital Signal Processing Systems, Design and Implementation, copyright 1999, by john Wiley & Sons Inc., reprint 2008. ISBN: 987-81-265-1098

K. Thriveni received Bachelor's Degree in Electronics and Communication Engineering from Vaagdevi Institute of Technology & Science and Master's degree in Embedded Systems & VLSI Design from Gouthami Institute of Technology & Management, JNTU Anantapur. She is currently working as an Assistant Professor in the Department of Electronics and Communication Engineering at BIT Institute of Technology, Hindupur. Her research interests include VLSI Design and Technology and Embedded Systems Design.