International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Sep 2024
https://ijgst.com.2024.v13.i2.pp1107-1114

# LUT BASED MULTIPLIER FOR SHORT WORD LENGTH DSP SYSTEMS

**DR J KALIAPPAN[1] CH SRINIVASA RAO [2] G SAI LAKSHMI [3] G THANUJA [4]**
[1]PROFESSOR, DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,
[2] ASSOC. PROFESSOR, DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,
[3]ASST.PROFESSOR, DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,
[4]ASST.PROFESSOR, DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,
[1,2,3,4] SRI MITTAPALLI COLLEGE OF ENGINEERING

***ABSTRACT*—**Continual need for high performance in compute bound scientific applications motivates the study of LUT Optimization. LUT Optimization replaces a complex expression with an access to precomputed LUT data containing the expression values. This results in faster expression evaluation and high performance gain. The LUT approach can result in significant improvement in terms of low latency implementation and less dynamic power consumption. Memory based computing structures are more regular than the multiply accumulate (MAC) structures.

## I. INTRODUCTION

A LONG with the progressive device scaling, semiconductor memory has become cheaper, faster, and more power-efficient. Moreover, according to the projections of the international technology roadmap for semiconductors [1], embedded memories will have dominating presence in the system on-chips (SoCs), which may exceed 90% of the total SOC content. It has also been found that the transistor packing density of memory components is not only higher but also increasing much faster than those of logic components. Apart from that, memory-based computing structures are more regular than the multiply–accumulate structures and offer many other advantages, e.g., greater potential for high-throughput and low-latency implementation and less dynamic power consumption. Memory-based computing is well suited for many digital signal processing (DSP) algorithms, which involve multiplication with a fixed set of coefficients. A conventional lookup-table (LUT)-based multiplier is shown in Fig. 1, where A is a fixed coefficient, and X is an input word to be multiplied with A. Assuming X to be a positive binary number of word length L, there can be 2L possible values of X, and accordingly, there can be 2L possible values of product C = A · X. Therefore, for memory-based multiplication, an LUT of 2L words, consisting of precomputed product values corresponding to all possible values of X, is conventionally used. The product word A · Xi is stored at the location Xi for $0 \leq Xi \leq 2L - 1$, such that if an L-bit binary value of Xi is used as the address for the LUT, then the corresponding product value A · Xi is available as its output. Several architectures have been reported in the literature for memory-based implementation of DSP algorithms involving orthogonal transforms and digital filters [2]–[8]. However, we do not find any significant work on LUT optimization for memory-based multiplication. Recently, we have presented a new approach to LUT design, where only the odd multiples of the fixed coefficient are required to be stored [9], which we have referred to as the odd-multiple-storage (OMS) scheme in this brief. In addition, we have shown that, by the anti symmetric product coding (APC) approach, the LUT size can also be reduced to half, where the product words are recoded as anti symmetric pairs [10]. The APC approach, although providing a reduction in LUT size by a factor of two, incorporates substantial

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Sep 2024
https://ijgst.com.2024.v13.i2.pp1107-1114

overhead of area and time to perform the two's complement operation of LUT output for sign modification and that of the input operand for input mapping. However, we find that when the APC approach is combined with the OMS technique, the two's complement operations could be very much simplified since the input address and LUT output could always be transformed into odd integers.1 However, the OMS technique in [9] cannot be combined with the APC scheme in [10], since the APC words generated according to [10] are odd numbers.

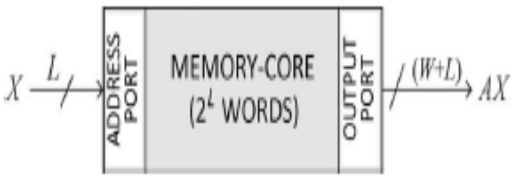



Fig:Conventional LUT-based multiplier

Moreover, the OMS scheme in [9] does not provide an efficient implementation when combined with the APC technique. In this brief, we therefore present a different form of APC and combined that with a modified form of the OMS scheme for efficient memory based multiplication. In the next section, we have discussed the modified APC and the combined OMS–APC approach. The implementation of combined OMS–APC scheme is described in Section III, and the design of the LUT-based multiplier for high input precision is discussed in Section IV. The synthesis results of the proposed multiplier and canonical-signed-digit (CSD)-based multipliers, along with the conclusion, are presented in Section V

**BINARY MULTIPLICATION:**

Multiplication in binary is similar to its decimal counterpart. Two numbers A and B can be multiplied by partial products: for each digit in B, the product of that digit in A is calculated and written on a new line, shifted leftward so that its rightmost digit lines up with the digit in B that was used. The sum of all these partial products gives the final result.

**MEMORY BASED MULTIPLICATION:**

The input-output relationship of an N-tap FIR filter in time-domain is given by

$$y(n) = h(0)\cdot x(n) + h(1)\cdot x(n-1) + h(2)\cdot x(n-2) + \cdots h(N-1)\cdot x(n-N+1)$$

where h(n), for n = 0,1,2,--------N-1, represent the filter coefficients x(n-i), while for i== 0,1,2,--------N-1, for x(n) , represent recent input samples y(n), and represents the current output sample. Memory-based multipliers can be implemented for signed as well as unsigned operands

**FIR FILTER ARCHITECTURE**

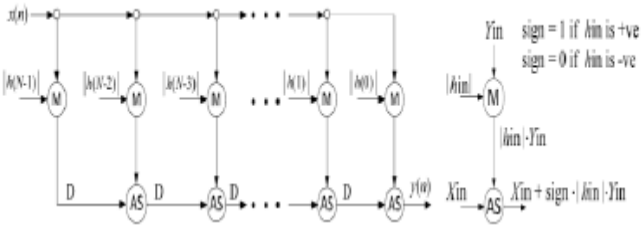



Fig : FIR Filter Architecture

**SOFT MULTIPLIERS:**

Soft multipliers are an extremely flexible alternative to using DSP blocks. Instead of implementing a combinatorial logic multiplier, they utilize a novel implementation based on a partial

look-up table (LUT) implementation of the multiplication operation, where the LUT is implemented in the memory blocks. Soft multipliers increase by a factor of between 2 and 15 the number of multipliers available. By downloading different coefficient LUTs, different configurations of multipliers and adders are generated.

Figure below shows simple Soft Multiplier implemented with a M512 (32*18) RAM block. The 5 bits width input data is driving the address bus of a memory and pointing to a LUT location that has the 18 bit result. The LUT covers all the multiplication combinations of 5 bits of input data with 13 bits coefficient.

## II. EXISTING SYSTEM:

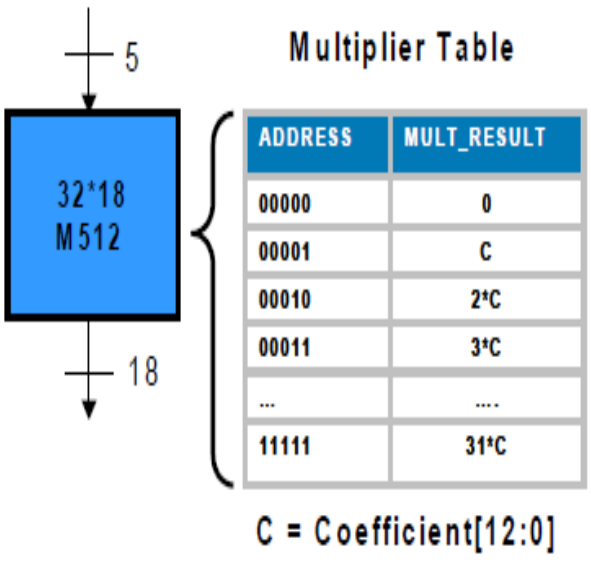
Several architectures have been reported in the literature for memory-based implementation of DSP algorithms involving orthogonal transforms and digital filters. However, we do not find any significant work on LUT optimization for memory-based multiplication. Recently, we have presented a new approach to LUT design, where only the odd multiples of the fixed coefficient are required to be stored , which we have referred to as the odd-multiple-storage (OMS) scheme. In addition, we have shown that, by the anti symmetric product coding (APC) approach, the LUT size can also be reduced to half, where the product words are recoded as ant symmetric pairs.

TABLE I
APC WORDS FOR DIFFERENT INPUT VALUES FOR $L = 5$

| Input, $X$ | product values | Input, $X$ | product values | address $x_3'x_2'x_1'x_0'$ | APC words |
|---|---|---|---|---|---|
| 0 0 0 0 1 | $A$ | 1 1 1 1 1 | $31A$ | 1 1 1 1 | $15A$ |
| 0 0 0 1 0 | $2A$ | 1 1 1 1 0 | $30A$ | 1 1 1 0 | $14A$ |
| 0 0 0 1 1 | $3A$ | 1 1 1 0 1 | $29A$ | 1 1 0 1 | $13A$ |
| 0 0 1 0 0 | $4A$ | 1 1 1 0 0 | $28A$ | 1 1 0 0 | $12A$ |
| 0 0 1 0 1 | $5A$ | 1 1 0 1 1 | $27A$ | 1 0 1 1 | $11A$ |
| 0 0 1 1 0 | $6A$ | 1 1 0 1 0 | $26A$ | 1 0 1 0 | $10A$ |
| 0 0 1 1 1 | $7A$ | 1 1 0 0 1 | $25A$ | 1 0 0 1 | $9A$ |
| 0 1 0 0 0 | $8A$ | 1 1 0 0 0 | $24A$ | 1 0 0 0 | $8A$ |
| 0 1 0 0 1 | $9A$ | 1 0 1 1 1 | $23A$ | 0 1 1 1 | $7A$ |
| 0 1 0 1 0 | $10A$ | 1 0 1 1 0 | $22A$ | 0 1 1 0 | $6A$ |
| 0 1 0 1 1 | $11A$ | 1 0 1 0 1 | $21A$ | 0 1 0 1 | $5A$ |
| 0 1 1 0 0 | $12A$ | 1 0 1 0 0 | $20A$ | 0 1 0 0 | $4A$ |
| 0 1 1 0 1 | $13A$ | 1 0 0 1 1 | $19A$ | 0 0 1 1 | $3A$ |
| 0 1 1 1 0 | $14A$ | 1 0 0 1 0 | $18A$ | 0 0 1 0 | $2A$ |
| 0 1 1 1 1 | $15A$ | 1 0 0 0 1 | $17A$ | 0 0 0 1 | $A$ |
| 1 0 0 0 0 | $16A$ | 1 0 0 0 0 | $16A$ | 0 0 0 0 | 0 |

For $X = (0\ 0\ 0\ 0\ 0)$, the encoded word to be stored is $16A$.

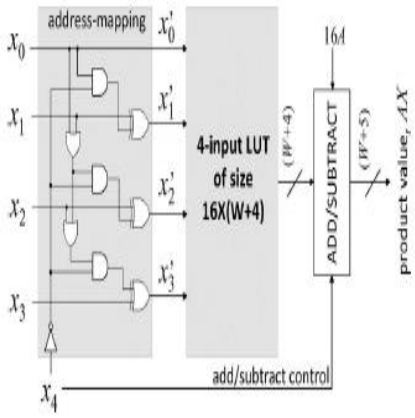
The structure and function of the LUT-based multiplier for $L = 5$ using the APC technique is shown in Fig. 2. It consists of a four-input LUT of 16 words to store the APC values of product words as given in the sixth column of Table I, except on the last row, where $2A$ is stored for input $X = (00000)$instead of storing a "0" for input $X = (10000)$. Besides, It consists of an address-mapping circuit and an add/subtract circuit. The address-mapping circuit generates the desired address (x3'x2'x1'x0') according to (2).



International Journal of Gender, Science and Technology
ISSN: 2040-0748

UGC Care Group I Journal.
Vol-13 Issue-02 Sep 2024

https://ijgst.com.2024.v13.i2.pp1107-1114




LUT-Based Multiplier For $L = 5$ Using The APC Technique.

## IMPLEMENTATION OF THE OPTIMIZED LUT USING MODIFIED OMS

The APC approach, although providing a reduction in LUT size by a factor of two, incorporates substantial overhead of area and time to perform the two's complement operation of LUT output for sign modification and that of the input operand for input mapping. However, we find that when the APC approach is combined with the OMS technique, the two's complement operations could be very much simplified since the input address and LUT output could always be transformed into odd integers.

1) A memory unit of $[(2L/2) + 1]$ words of $(W + L)$-bit width is used to store the product values, where the first$(2L/2)$ words are odd multiples of $A$, and the last word is zero.

2) A barrel shifter for producing a maximum of $(L - 1)$ left shifts is used to derive all the even multiples of $A$.

3) The $L$-bit input word is mapped to the $(L - 1)$-bit address of the LUT by an address encoder, and control bits for the barrel shifter are derived by a control circuit.

TABLE II
OMS-BASED DESIGN OF THE LUT OF APC WORDS FOR $L = 5$

| input $X'$ $x_3'x_2'x_1'x_0'$ | product value | # of shifts | shifted input, $X''$ | stored APC word | address $d_3d_2d_1d_0$ |
|---|---|---|---|---|---|
| 0 0 0 1 | $A$ | 0 | 0 0 0 1 | $P0 = A$ | 0 0 0 0 |
| 0 0 1 0 | $2 \times A$ | 1 | | | |
| 0 1 0 0 | $4 \times A$ | 2 | | | |
| 1 0 0 0 | $8 \times A$ | 3 | | | |
| 0 0 1 1 | $3A$ | 0 | 0 0 1 1 | $P1 = 3A$ | 0 0 0 1 |
| 0 1 1 0 | $2 \times 3A$ | 1 | | | |
| 1 1 0 0 | $4 \times 3A$ | 2 | | | |
| 0 1 0 1 | $5A$ | 0 | 0 1 0 1 | $P2 = 5A$ | 0 0 1 0 |
| 1 0 1 0 | $2 \times 5A$ | 1 | | | |
| 0 1 1 1 | $7A$ | 0 | 0 1 1 1 | $P3 = 7A$ | 0 0 1 1 |
| 1 1 1 0 | $2 \times 7A$ | 1 | | | |
| 1 0 0 1 | $9A$ | 0 | 1 0 0 1 | $P4 = 9A$ | 0 1 0 0 |
| 1 0 1 1 | $11A$ | 0 | 1 0 1 1 | $P5 = 11A$ | 0 1 0 1 |
| 1 1 0 1 | $13A$ | 0 | 1 1 0 1 | $P6 = 13A$ | 0 1 1 0 |
| 1 1 1 1 | $15A$ | 0 | 1 1 1 1 | $P7 = 15A$ | 0 1 1 1 |

The product values and encoded words for input words $X$ = (00000) and (10000) are separately shown in Table III. For $X$ = (00000), the desired encoded word $16A$ is derived by 3-bit left shifts of $2A$ [stored at address (1000)]. For $X$ =(10000), the APC word "0" is derived by resetting the LUT output, by an active-high RESET signal given by

$$\text{RESET} = (x0 + x1 + x2 + x3)'x4$$

TABLE III
PRODUCTS AND ENCODED WORDS FOR $X$ = (00000) AND (10000)

| input $X$ $x_4x_3x_2x_1x_0$ | product values | encoded word | stored values | # of shifts | address $d_3d_2d_1d_0$ |
|---|---|---|---|---|---|
| 1 0 0 0 0 | $16A$ | 0 | - - - | -- | - - - |
| 0 0 0 0 0 | 0 | $16A$ | $2A$ | 3 | 1 0 0 0 |

$$X'' = \begin{cases} Y_L, & \text{if } x_4 = 1 \\ Y_L', & \text{if } x_4 = 0 \end{cases}$$

International Journal of Gender, Science and Technology
ISSN: 2040-0748

UGC Care Group I Journal.
Vol-13 Issue-02 Sep 2024

https://ijgst.com.2024.v13.i2.pp1107-1114

### III. PROPOSED SYSTEM:

The proposed APC–OMS combined design of the LUT for $L = 5$ and for any coefficient width $W$ is shown in below Fig. It consists of an LUT of nine words of $(W + 4)$-bit width, a four-to-nine-line address decoder, a barrel shifter, an address generation circuit, and a control circuit for generating the RESET signal and control word $(s1s0)$ for the barrel shifter.

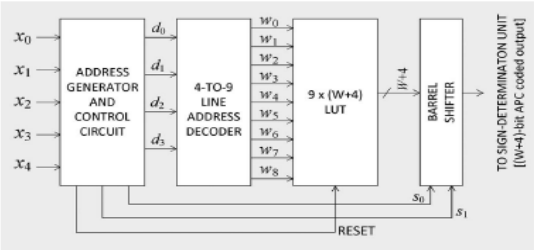



Fig : Proposed APC–OMS combined LUT design for the multiplication of
$W$-bit fixed coefficient $A$ with 5-bit input $X$

The pre computed values of $A \times (2i + 1)$ are stored as $Pi$, for $i = 0, 1, 2, \ldots, 7$, at the eight consecutive locations of the memory array, as specified in Table II, while $2A$ is stored for input $X$ = (00000) at LUT address "1000," as specified in Table III. The decoder takes the 4-bit address from the address generator and generates nine word-select signals i.e., $\{wi,$ for $0 \le i \le 8\}$, to select the referenced word from the LUT. The 4-to-9-line decoder is a simple modification of 3-to-8-line decoder, as shown in below Fig(a). The control bits $s0$ and $s1$to be used by the barrel shifter to

produce the desired number of shifts of the LUT output are generated by the control circuit according to the relations.

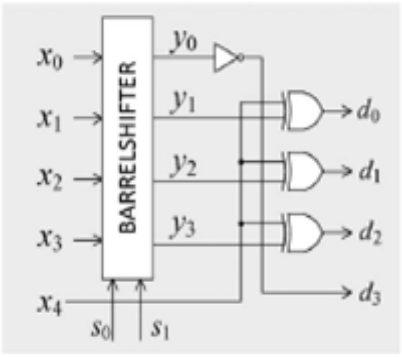



**Fig: Address Generation Unit**

Note that $(s1s0)$ is a 2-bit binary equivalent of the required number of shifts specified in Tables II and III. The RESET signal given by (4) can alternatively be generated as ($d3$ AND $x4$).The control circuit to generate the control word and RESET is shown in below Fig (b). The address-generator circuit receives the 5-bit input operand $X$ and maps that onto the 4-bit address word

$(d3d2d1d0)$, according to (5) and (6).

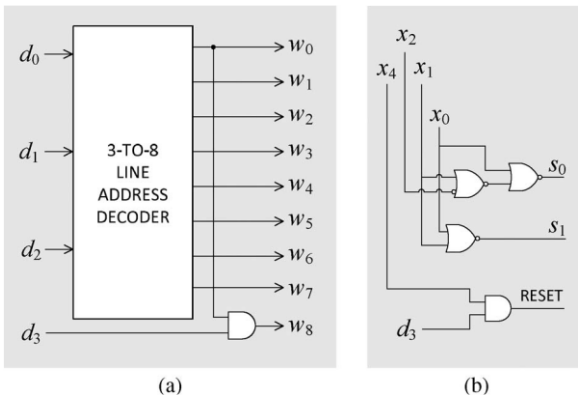



Fig: a) four-to-nine line address decoder
Fig: b) control circuit generation of S0,S1&RESET

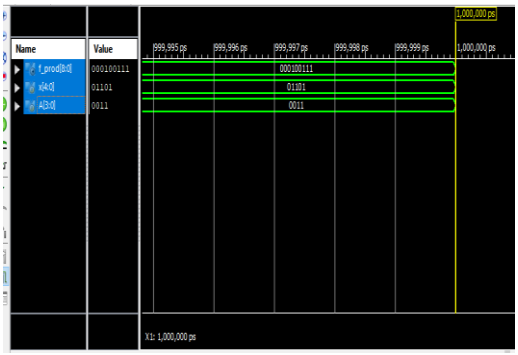
International Journal of Gender, Science and Technology
ISSN: 2040-0748

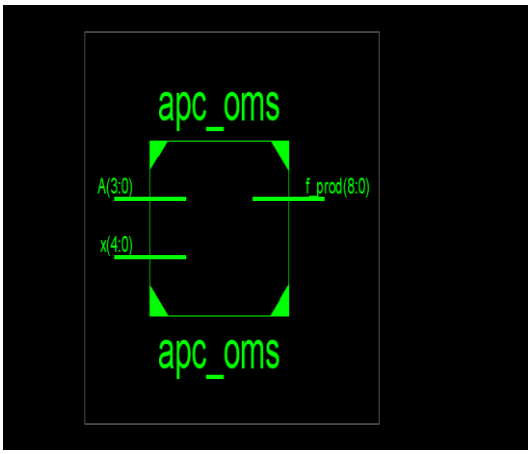
UGC Care Group I Journal.
Vol-13 Issue-02 Sep 2024

https://ijgst.com.2024.v13.i2.pp1107-1114

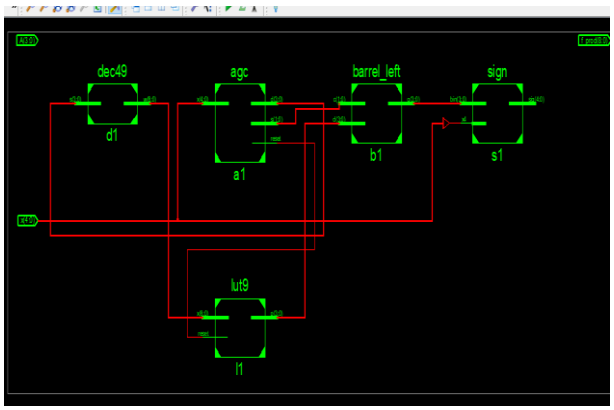
## IV. RESULTS

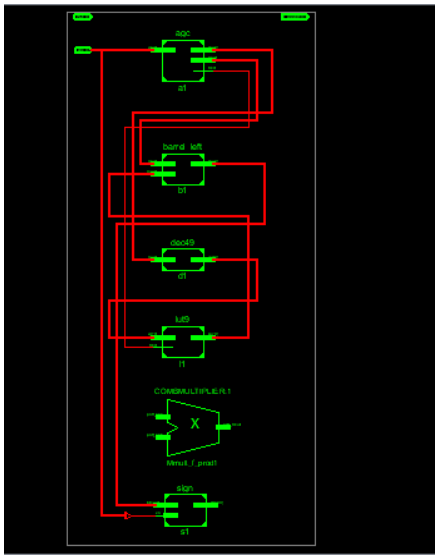
### SIMULATION RESULT




### RTL SCHEMATIC



### DESIGN SUMMARY
### FOR EXISTING SYSTEM

International Journal of Gender, Science and Technology
ISSN: 2040-0748

UGC Care Group I Journal.
Vol-13 Issue-02 Sep 2024

https://ijgst.com.2024.v13.i2.pp1107-1114

| top_APC Project Status | | | |
|---|---|---|---|
| Project File: | akhil.xise | Parser Errors: | No Errors |
| Module Name: | top_APC | Implementation State: | Synthesized |
| Target Device: | xc3s500e-5fg320 | • Errors: | No Errors |
| Product Version: | ISE 13.2 | • Warnings: | No Warnings |
| Design Goal: | Balanced | • Routing Results: | |
| Design Strategy: | Xilinx Default (unlocked) | • Timing Constraints: | |
| Environment: | System Settings | • Final Timing Score: | |

| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| Logic Utilization | Used | Available | Utilization |
| Number of Slices | 23 | 4656 | 0% |
| Number of 4 input LUTs | 40 | 9312 | 0% |
| Number of bonded IOBs | 18 | 232 | 7% |

**FOR PROPOSED SYSTEM**

| apc_oms Project Status | | | |
|---|---|---|---|
| Project File: | akhil2.xise | Parser Errors: | No Errors |
| Module Name: | apc_oms | Implementation State: | Synthesized |
| Target Device: | xc3s500e-5fg320 | • Errors: | No Errors |
| Product Version: | ISE 13.2 | • Warnings: | 2 Warnings (2 new) |
| Design Goal: | Balanced | • Routing Results: | |
| Design Strategy: | Xilinx Default (unlocked) | • Timing Constraints: | |
| Environment: | System Settings | • Final Timing Score: | |

| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| Logic Utilization | Used | Available | Utilization |
| Number of Slices | 11 | 4656 | 0% |
| Number of Slice Flip Flops | 4 | 9312 | 0% |
| Number of 4 input LUTs | 20 | 9312 | 0% |
| Number of bonded IOBs | 18 | 232 | 7% |
| Number of MULT18X18SIOs | 1 | 20 | 5% |

## IV. CONCLUSION

The proposed LUT multipliers for word size L = W = 5 and 6 bits are coded in verilog and synthesized in Xilinx ISE 10.1i. Simulation Part is done in Modelsim 6.4b, where the LUTs are implemented as arrays of constants, and additions are implemented by the Wallace tree and ripple carry array. The CSD-based multipliers having the same addition schemes are also synthesized with the same technology library. We have shown the possibility of using LUT based multipliers to implement the constant multiplication for DSP applications.

## V. FUTURESCOPE

FPGAs and other programmable logic arrays are highly configurable. Further work could still be carried out to derive such modified OMS based LUTs for higher input sizes with different decomposition forms. Other parallel and pipelined addition schemes for suitable area-delay tradeoffs. The LUT multipliers for word size L = W = 8, 16, and 32 bits can be coded and synthesizing using Xilinx ISE 12.2i. For the Simulation Part we are going to use Modelsim 6.4b for More Less Area and Less Multiplication Time.

## VI. BIBLIOGRAPHY

1. International Technology Roadmap for Semiconductors. [Online].Available: http://public.itrs.net/
2. J.-I. Guo, C.-M. Liu, and C.-W. Jen, “The efficient memory-based VLSI array design for DFT and DCT,” *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.* vol. 39, no. 10, pp. 723–733, Oct. 1992.
3. H.-R. Lee, C.-W. Jen, and C.-M. Liu, “On the design automation of the memory-based VLSI architectures for FIR filters,” *IEEE Trans. Consum. Electron.*, vol. 39, no. 3, pp. 619–629, Aug. 1993.
4. D. F. Chiper, M. N. S. Swamy, M. O. Ahmad, and T. Stouraitis, “A systolicarray architecture for the discrete sine transform,” *IEEE Trans. SignalProcess.*, vol. 50, no. 9, pp. 2347–2354, Sep. 2002.
5. H.-C. Chen, J.-I. Guo, T.-S. Chang, and C.-W. Jen, “A memory-efficientrealization of cyclic

convolution and its application to discrete cosinetransform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3,pp. 445–453, Mar. 2005.

6. D. F. Chiper, M. N. S. Swamy, M. O. Ahmad, and T. Stouraitis, "Systolicalgorithms and a memory-based design approach for a unified architecturefor the computation of DCT/DST/IDCT/IDST," *IEEE Trans. CircuitsSyst. I, Reg. Papers*, vol. 52, no. 6, pp. 1125–1137, Jun. 2005.
7. P. K. Meher, "Systolic designs for DCT using a low-complexity concurrent convolutional formulation," *IEEE Trans. Circuits Syst. Video Technol.*,vol. 16, no. 9, pp. 1041–1050, Sep. 2006.
8. P. K. Meher, "Memory-based hardware for resource-constrained digitalsignal processing systems," in *Proc. 6th Int. Conf. ICICS*, Dec. 2007,pp. 1–4.
9. P. K. Meher, "New approach to LUT implementation and accumulation for memory-based multiplication," in *Proc. IEEE ISCAS*, May 2009,pp. 453–456.
10. P. K. Meher, "New look-up-table optimizations for memory-based multiplication," in Proc. ISIC, Dec. 2009, pp. 663–666.
11. A.K Sharma, Advanced Semiconductor Memories: Architectures, Design and Applications, Piscataway, NJ: IEEE Press, 2003.