ENGINEERING IN ADVANCED
RESEARCHSCIENCE AND
TECHNOLOGY

ISSN 2352-8648
Vol.01, Issue.03
April-2022
Pages: -740-751

APC AND OMS BASED LUT DESIGN FOR HIGH THROUGHPUT DIF IMPLEMENTATION

[1]CHINTAPALLI HIMABINDU¸ [2]MANDAVILLI SRI HARI
[1]M. Tech, Dept. of ECE, Kakinada Institute Of Engineering and Technology for Women, Kakinada, A.P
[2]Assistant Professor, Dept. of ECE, KIET group of institutions, Kakinada, A.P

ABSTRACT: Digital Intermediate Frequency (DIF) is the vital innovation in digit filters. DIF filter is depend on filter tap, it means that the high tap in system complexity is high and low tap in system complexity is low. In this concept, high tap and low system complexity architecture is used with multi-rate support. In direct plan strategy for FIR channels created there is a command over the perfection of the size reaction, But in the event that there will be an additional boundary for controlling the size reaction we can accomplish less mistake in the passband and stopband. In this way, in this work we have built up a FIR channel that has extra authority over the size reaction and have less passband and stopband blunder. A new approach to LUT design is presented, where only the odd multiples of the fixed coefficient are required to be stored, which is referred to as the OMS. In addition, by the APC approach, the LUT size can also be reduced to half, where the product words are recoded as anti-symmetric pairs. 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, it is found 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. However, the OMS technique cannot be combined with the APC scheme, since the APC words generated according to are odd numbers. Moreover, the OMS scheme in does not provide an efficient implementation when combined with the APC technique. In this brief, a different form of APC is presented combined with a modified form of the OMS scheme for efficient memory based multiplication.

Keywords: Digital Intermediate Frequency, Look Up Table, Anti Symmetric Code, Odd multiple Storage, Finite Impulse Response.

INTRODUCTION: FIR DIGITAL filters find extensive applications in mobile communication systems for applications such as channelization, channel equalization, matched filtering, and pulseshaping, due to

Copyright @ 2022**ijearst. All rights reserved.**

their absolute stability and linear phase properties. The filters employed in mobile systems must be realizedto consume less power and operate at high speed. Recently,with the advent of software defined radio (SDR) technology,finite impulse response (FIR) filter research has been focusedon reconfigurable realizations. The fundamental idea of anSDR is to replace most of the analog signal processing in thetransceivers with digital signal processing in order to providethe advantage of flexibility through reconfiguration. This will enable different air-interfaces to be implemented on a single generic hardware platform to support multi standard wireless communications [1]. Wideband receivers in SDR must be realized to meet the stringent specifications of low power consumption and high speed. Re-configurability of the receiver to work with different wireless communication standards is another key requirement in an SDR. The most computationally intensive part of an SDR receiver is the channelizer since it operates at the highest sampling rate [2]. It extracts multiple narrow band channels from a wideband signal using a bank of FIR filters, called channel filters. Using poly phase filter structure, decimation can be done prior to channel filtering so that the channel filters need to operate only at relatively low sampling rates. This can relax the speed of operation ofthe filters to a good extent [22]. However due to the stringent adjacent channel attenuation specifications of wireless communication standards, higher order filters are required for channelization and consequently the complexity and power consumption of the receiver will be high. As the ultimate aim of the future multi-standard wireless communication receiver is to realize its functionalities in mobile handsets, where its full utilization is possible, low power and low area implementation of FIR channel filters is inevitable. In [37], the filter multiplications are done via state machines in an iterative shift and add component and as a result of this there is huge savings in area. For lower order filters, the approach in [37] offers good trade-off between speed and area. But in general, the channel filters in wireless communication receivers need to be of high order to achieve sharp transition band and low adjacent channel attenuation requirements. For such applications, the approach in [37] results in low speed of operation. The complexity of FIR filters is dominated by the complexityof coefficient multipliers. It is well known that the common sub expression elimination (CSE) methods based on canonical signed digit (CSD) coefficients produce low complexity FIR filter coefficient multipliers [3]. The goal of CSE is to identify multiple occurrences of identical bit patterns that are present in the CSD representation of coefficients, and eliminate these redundant multiplications. A modification of the 2-bit CSE technique in [3] for identifying the proper patterns for elimination of redundant computations and to maximize the optimization impact was proposed in [4]. In [5],the technique in [3] was modified to minimize the logic depth(LD) (LD is defined as the number of adder-steps in a maximal path of decomposed multiplications [27]) and thus to improve the speed of operation.

Copyright @ 2022ijearst. All rights reserved.

LITERATURE SURVEY: The research paper on the design of FIR filters are published in various journals and presented in many conferences. Here the paper selected describes the design of FIR filters using VHDL or Verilog language.Some of the paper represents the modular design approach of the FIR filters and which is implemented in spartan-3E FPGA/Xilinx Virtex-5 FPGA.The evaluation result shows good area/power efficiency and flexibility by using different architectures for application. .Most papers have used microprogrammed FIR filters design approach .Abdullah A.Aljuffri, Aiman S. Badawai , Mohammad S.Bensaleh, Abdulfattah M.Odeid and SayedManzoor Qasim [1] in paper entitled "FPGA implementation of scalable micro programmed FIR filter architectures using Wallace tree and Vedic multipliers". In this paper used Wallace Tree and Vedic multipliers for implementation of 8-tap and 16-tap sequential and parallel micro programmed FIR filters architectures The designs are realized using Xilinx virtex-5 FPGA. Synplify pro tool used for synthesis, translation, mapping and place and route process and Reports are generated by CAD tool. Performance analyze base on parameter such as minimum period, slice LUTs and maximum operating frequency. The sequential FIR filters architecture designed using Wallace Tree multiplier seems to be more efficient as compared to Vedic multipliers. For 8-tap FIR filter using Wallace Tree have minimum period 11.448 ns and maximum operating frequency 87.4 MHz And for 16-tap FIR filter using Wallace Tree have minimum period 10.491 ns and maximum operating frequency 85.3 MHz .A. Aljuffri. M. M. AlNahdi, A.A.Hemaid , O. A. Alshaalan, M. S. BenSaleh, A.M. Obeid and S. M. Qasim [2], in paper entitled, "ASIC realization and performance evaluation of scalable micro-programmed FIR filter architectures using Wallace tree and Vedic multiplier". In this paper, Wallace tree and Vedic multiplier are used for efficient realization of 8-tap and 16-tap sequential and parallel scalable micro-programmed FIR filter architectures. The designs of FIR filter are coded in VHDL. Lfoundary 150nm standard-cell based technology is used for the hardware realization of the proposed designs in ASIC. Synopsys Design Compiler is used for thegate-level synthesis. Analyze the performance based on area, Slice LUTs and critical path delays. Wallace tree multiplier using CSA (Carry Skip Adder) has minimum area and delay while Vedic using KSA (Kogge-Stone Adder) has maximum area and delay.

FINITE IMPULSE RESPONSE: A finite impulse response (FIR) filter is a filter structure that can be used to implement almost any sort of frequency response digitally. An FIR filter is usually implemented by using a series of delays, multipliers, and adders to create the filter's output. Below figure shows the basic block diagram for an FIR filter of length N. The delays result in operating on prior input samples. The h_k values are the coefficients used for multiplication, so that the output at time n is the summation of all the delayed samples multiplied by the appropriate coefficients.

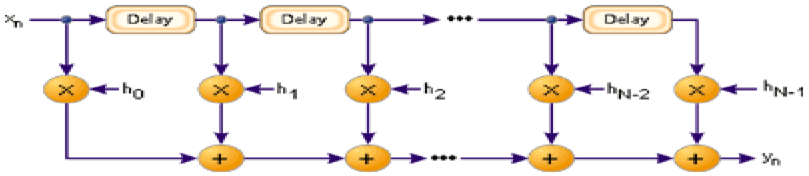


Figure1. The logical structure of an FIR filter

The process of selecting the filter's length and coefficients is called filter design. The goal is to set those parameters such that certain desired stopband and passband parameters will result from running the filter. Most engineers utilize a program such as MATLAB to do their filter design. But whatever tool is used, the results of the design effort should be the same:

- A frequency response plot, like the one shown in Figure 1, which verifies that the filter meets the desired specifications, including ripple and transition bandwidth.
- The filter's length and coefficients.

The longer the filter (more taps), the more finely the response can be tuned.

With the length, N, and coefficients, float h[N] = { ... }, decided upon, the implementation of the FIR filter is fairly straightforward. Listing 1 shows how it could be done in C. Running this code on a processor with a multiply-and-accumulate instruction (and a compiler that knows how to use it) is essential to achieving a large number of taps.

PROPOSED METHOD:

A new approach to LUT design is presented, where only the odd multiples of the fixed coefficient are required to be stored, which is referred to as the OMS. In addition, by the APC approach, the LUT size can also be reduced to half, where the product words are recoded as anti-symmetric pairs.

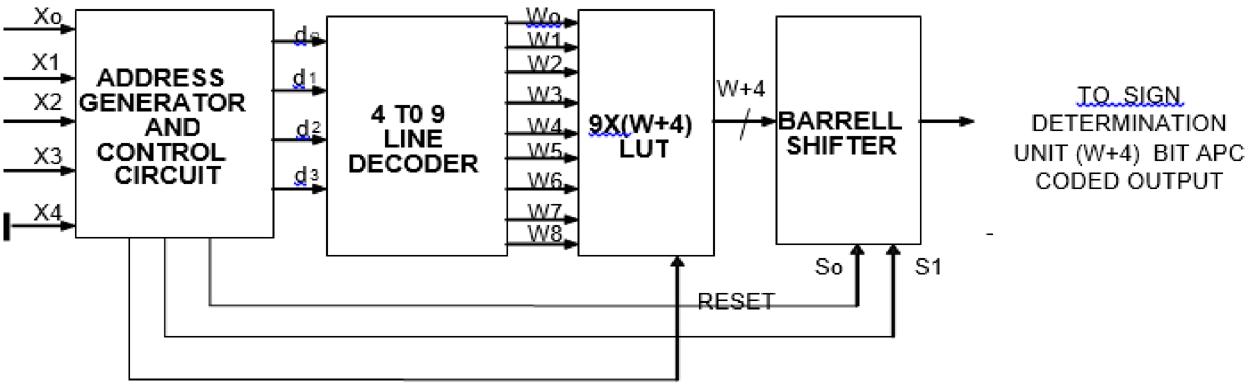


Fig2: Proposed LUT based multiplier

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

Copyright @ 2022ijearst. All rights reserved.

and that of the input operand for input mapping. However, it is found 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 cannot be combined with the APC scheme, since the APC words generated according to are odd numbers. Moreover, the OMS scheme in does not provide an efficient implementation when combined with the APC technique. In this brief, a different form of APC is presented combined with a modified form of the OMS scheme for efficient memory based multiplication.

Proposed LUT APC Part: The structure and function of the LUT-based multiplier for $L = 5$ using the APC technique is shown in Fig.3.2 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'). A straightforward implementation of address mapping can be done by $X'L$ using $x4$ as the control bit. The address-mapping circuit, however, can be optimized to be realized by three XOR gates, three AND gates, two OR gates, and a NOT gate, as shown in Fig. 3.2 Note that the RESET can be generated by a control circuit (not shown in this figure) .The output of the LUT is added with or subtracted from $16A$, for $x4 = 1$ or 0, respectively, by the add/subtract cell. Hence, $x4$ is used as the control for the add/subtract cell.

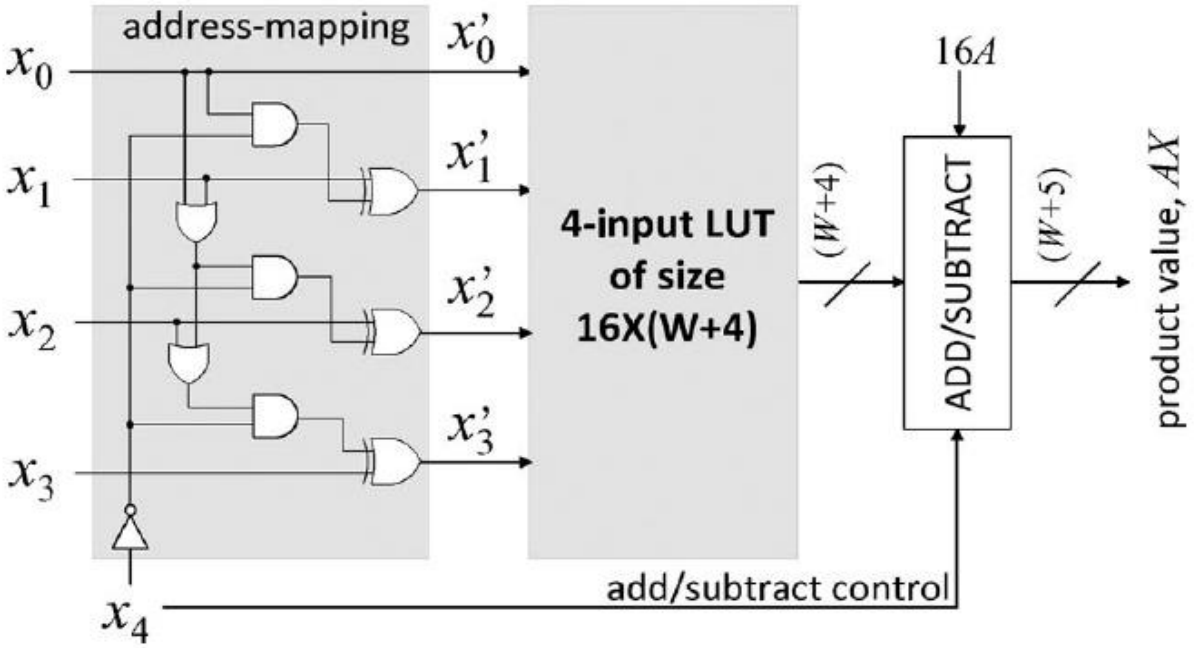


Figure3 Proposed APC Part

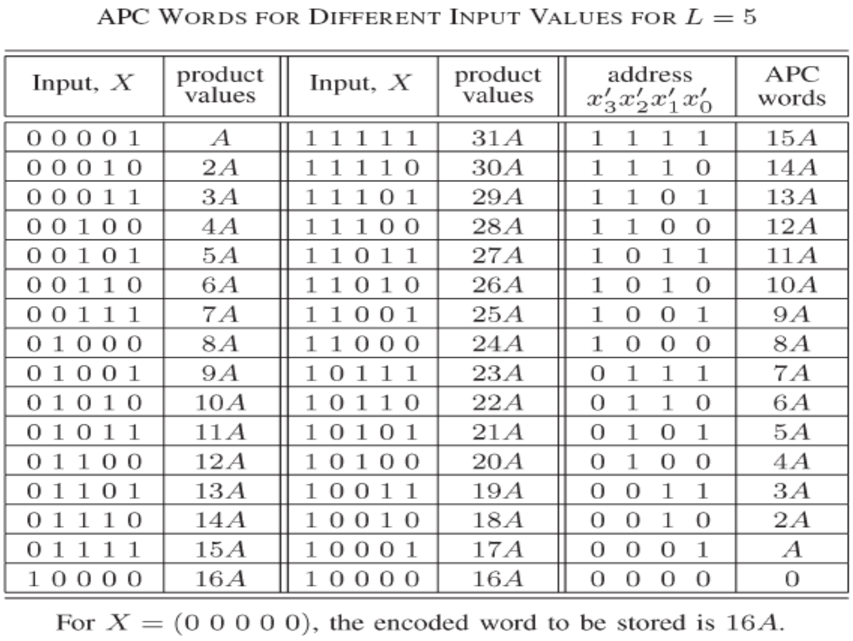
Copyright @ 2022ijearst. All rights reserved.

For simplicity of presentation, it is assumed both X and A to be positive integers. The product words for different values of X for L = 5 are shown in Table I. It may be observed in this table that the input word X on the first column of each row is the two's complement of that on the third column of the same row. In addition, the sum of product values corresponding to these two input values on the same row is 32A. LUT based multiplier for L=5 using the APC technique

W = Width of A

L = Width of X

Table 1: Stored APC Words



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

Input, X	product values	Input, X	product values	address $x'_3x'_2x'_1x'_0$	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 $16A$.

16 x (W+4) → 16 Locations and each location having (W+4) bits.

. Let the product values on the second and fourth columns of a row be u and v, respectively. Since one can write

u = [(u + v)/2 − (v − u)/2] and

v = [(u + v)/2 + (v − u)/2], for (u + v) = 32A,

U=16A+ [(V-U)/2] V=16A-[(V-U)/2]

Copyright @ 2022ijearst. All rights reserved.

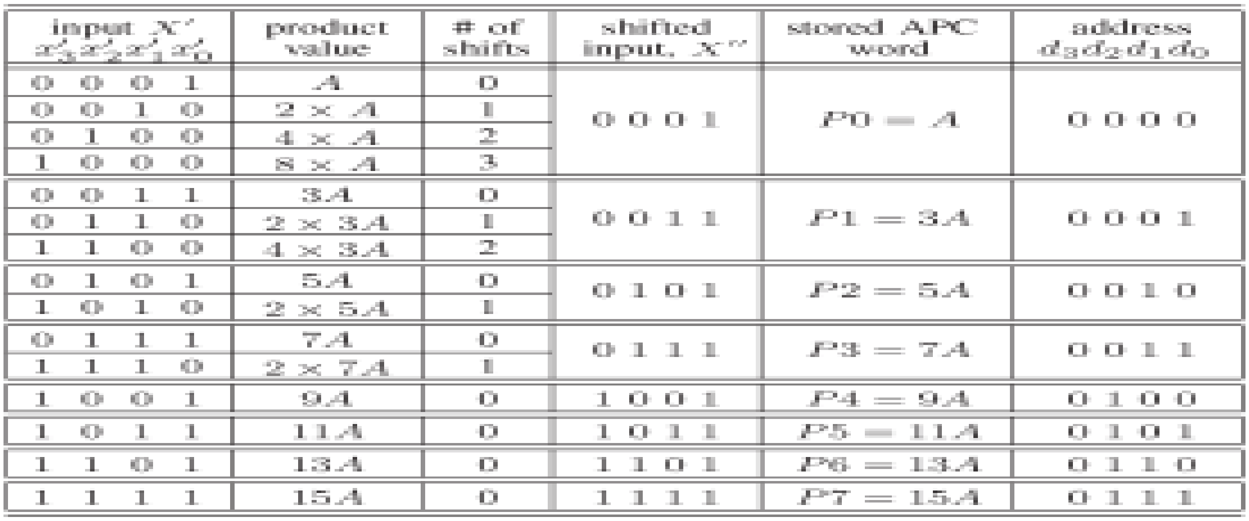
The product values on the second and fourth columns of Table I therefore have a negative mirror symmetry. This behavior of the product words can be used to reduce the LUT size, where, instead of storing u and v, only [(v − u)/2] is stored for a pair of input on a given row. The 4-bit LUT addresses and corresponding coded words are listed on the fifth and sixth columns of the table, respectively. Since the representation of the product is derived from the antisymmetric behavior of the products, we can name it as antisymmetric product code. The 4-bit address X_ = (x_3,x_2,x_1,x_0) of the APC word is given by

$$X' \left\{ \begin{array}{l} X_L, \text{if } x4=1 \\ X'_L, \text{if } x4=0 \end{array} \right.$$

Proposed APC-OMS Part : For the multiplication of any binary word X of size L, with a fixed coefficient A, instead of storing all the 2L possible values of C = A · X, only (2L/2) words corresponding to the odd multiples of A may be stored in the LUT, while all the even multiples of A could be derived by left-shift operations of one of those odd multiples. Based on the above assumptions, the LUT for the multiplication of an L-bit input with a W-bit coefficient could be designed by the following strategy.

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.

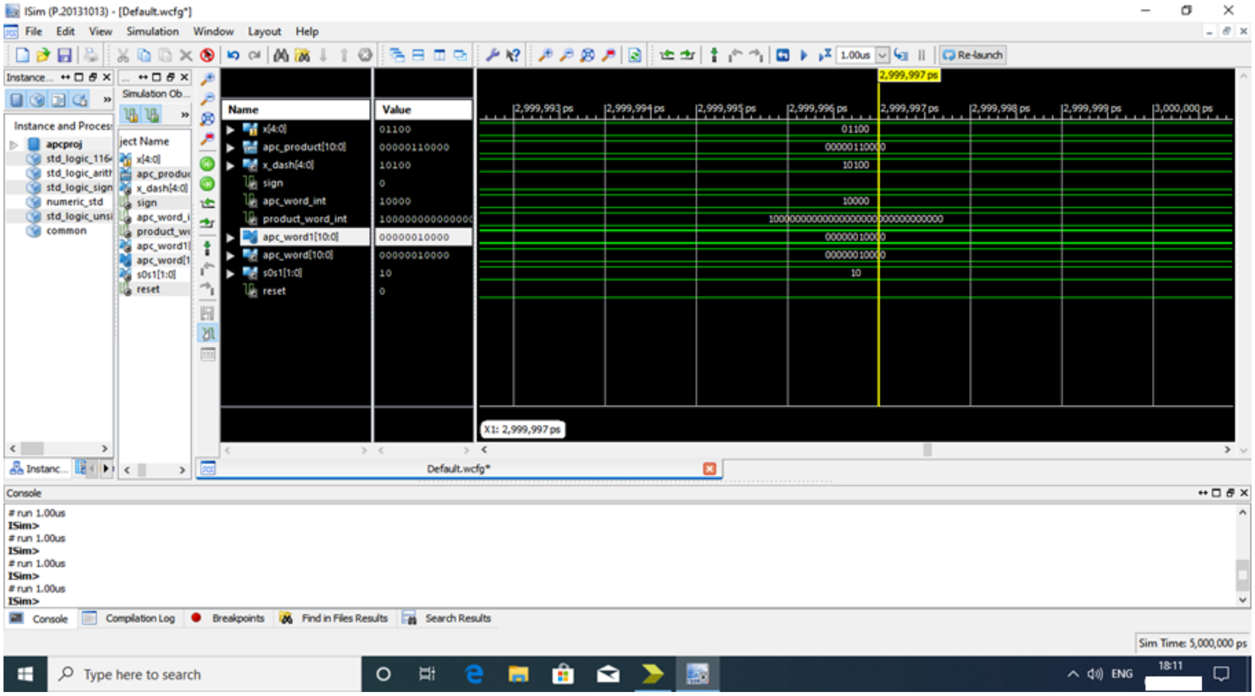
Table2 Stored APC-OMS Words



input X' $x'_3x'_2x'_1x'_0$	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

Copyright @ 2022ijearst. All rights reserved.

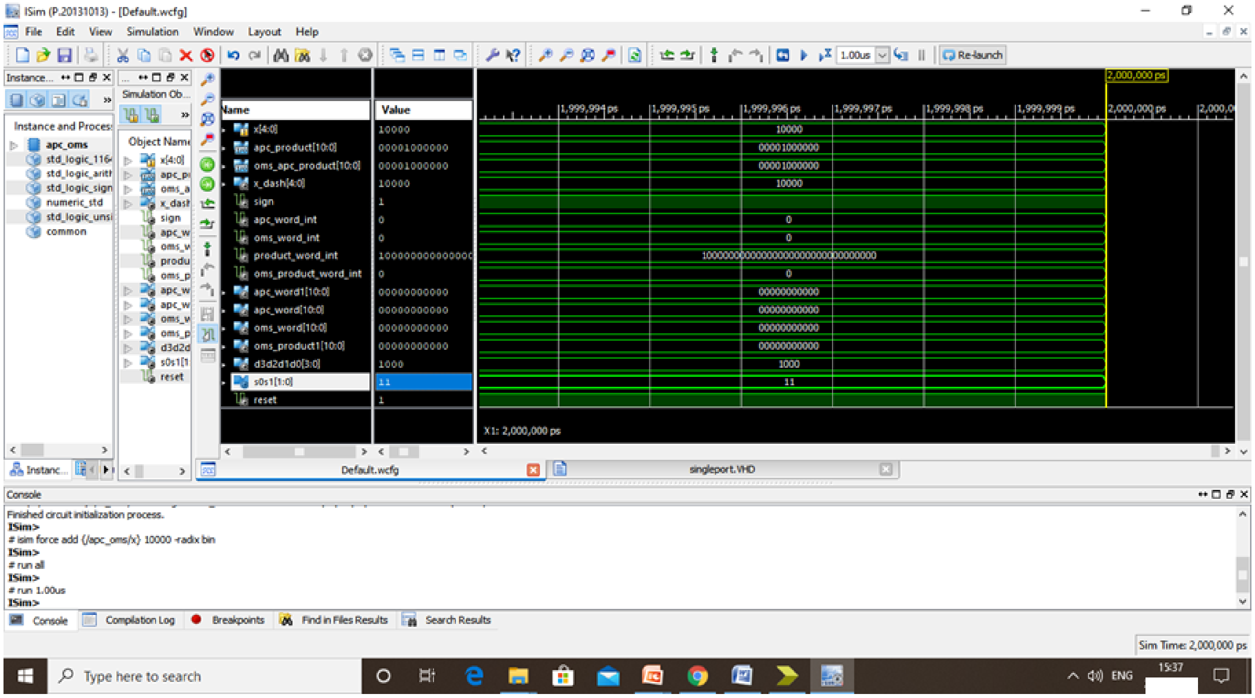
RESULT:



Name	Value
x[4:0]	01100
apc_product[10:0]	00000110000
x_dash[4:0]	10100
sign	0
apc_word_int	10000
product_word_int	100000000000000
apc_word1[10:0]	00000010000
apc_word[10:0]	00000010000
s0s1[1:0]	10
reset	0

```
# run 1.00us
ISim>
# run 1.00us
ISim>
# run 1.00us
ISim>
# run 1.00us
ISim>
```

Fig: Simulation result with 1100 input



Name	Value
x[4:0]	10000
apc_product[10:0]	00001000000
oms_apc_product[10:0]	00001000000
x_dash[4:0]	10000
sign	1
apc_word_int	0
oms_word_int	0
product_word_int	100000000000000
oms_product_word_int	0
apc_word1[10:0]	00000000000
apc_word[10:0]	00000000000
oms_word[10:0]	00000000000
oms_product1[10:0]	00000000000
d3d2d1d0[3:0]	1000
s0s1[1:0]	11
reset	1

```
Finished circuit initialization process.
ISim>
# isim force add {/apc_oms/x} 10000 -radix bin
ISim>
# run all
ISim>
# run 1.00us
ISim>
```

Fig: Simulation result with 10000 input

Copyright @ 2022**ijearst. All rights reserved.**

CONCLUSION:

In this paper, we have explored the possibility of realization of block FIR filters in transpose form configuration for area delay efficient realization of both fixed and reconfigurable applications. A generalized block formulation is presented for transpose form block FIR filter, and based on that we have derived transpose form block filter for reconfigurable applications. This method presented a scheme to identify the MCM blocks for APC and OMS in the proposed block FIR filter for fixed coefficients to reduce the computational complexity.

FUTURE SCOPE:

1. Research can be extended in design of FIR filter using various optimization techniques ACO, PSO etc. 2. In further work FIR filters can be design using evolutionary algorithms etc. we can design according to that recent technique and low power processor can be designed

REFERENCES:

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, N.J., USA: Prentice Hall, 1993.

[2] A. Sibille, C. Oestges and A. Zanella, *MIMO: From Theory to Implementation*, New York, NY, USA: Academic, 2010.

[3] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu, *Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro- Magnetic Disturbances*, New York, NY, USA: Springer Verlag, 2010.

[4] M. Nicolaidis, “Design for soft error mitigation,” *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 405–418, Sep. 2005.

[5] C. L. Chen and M. Y. Hsiao, “Error-correcting codes for semiconductor memory applications: A state-of-the-art review,” *IBM J. Res. Develop.*, vol. 28, no. 2, pp. 124–134, Mar. 1984.

[6] A. Reddy and P. Banarjee “Algorithm-based fault detection for signal processing applications,” *IEEE Trans. Comput.*, vol. 39, no. 10, pp. 1304–1308, Oct. 1990.

[7] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, “Totally fault tolerant RNS based FIR filters,” in *Proc. IEEE IOLTS*, 2008, pp. 192–194.

[8] Z. Gao, W. Yang, X. Chen, M. Zhao and J. Wang, “Fault missing rate analysis of the arithmetic residue codes based fault-tolerant FIR filter design,” in *Proc. IEEE IOLTS*, 2012, pp. 130–133.

[9] B. Shim and N. Shanbhag, “Energy-efficient soft error-tolerant digital signal processing,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14,no. 4, pp. 336–348, Apr. 2006.

Copyright @ 2022**ijearst. All rights reserved.**

[10] Y.-H. Huang, "High-efficiency soft-error-tolerant digital signal processing using fine-grain subword-detection processing," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, no 2, pp. 291–304, Feb. 2010.

[11] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "Structural DMR: A technique for implementation of soft-error-tolerant FIR filters," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 58, no. 8, pp. 512–516, Aug. 2011.

[12] P. Reviriego, S. Pontarelli, C. Bleakley and J. A. Maestro, "Area efficient concurrent error detection and correction for parallel filters," *IET Electron. Lett.*, vol. 48, no 20, pp. 1258–1260, Sep. 2012.

[13] Z. Gao *et al.*, "Fault tolerant parallel filters based on error correction codes," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 2, pp. 384–387, Feb. 2015.

[14] R. W. Hamming, "Error correcting and error detecting codes," *Bell Sys. Tech. J.*, vol. 29, pp. 147–160, Apr. 1950.

Copyright @ 2022**ijearst. All rights reserved.**