Design of Higher Order FIR Filter for Different Techniques of Multiplication

Gaurav Gautam[1], Prashant K. Shah[2]
[1]*M.Tech-VLSI &Embedded System, Department of Electronics Engineering, SVNIT Surat, India*
[2]*Associate Professor, Department of Electronics Engineering, SVNIT Surat, India*

***Abstract*- Today's DSP systems are well suited to VLSI implementation and if implemented using VLSI technologies they are often economically viable or technically feasible. The focus of this paper is to the design an FIR filter with efficient VLSI architecture to reduce the power consumption and the hardware complexity to optimize the filter area, delay and power. Basically optimization of"filter'area, delay and power is done by using add & shift method for.multiplication, but here power dissipation of'the'filter will increase. Complexity of the filter can be reduced by representing the coefficients in canonical signed digit'(CSD) representation as it is more efficient than the traditional"binary representation. In this paper, we will see different multiplication techniques for filter design such as, add and shift method, Vedic multiplier, Booth Multiplier and Wallace tree'(WT) multiplier for the multiplication of'filter coefficients with filter input. MATLAB is used for designing of Finite Impulse Response (FIR) filter using the equiripple'method and the same filter is synthesized on Xilinx Spartan3E target field-programmable gate array(FPGA) device using Hardware Description Language(HDL) and the total on-chip'power is calculated in Vivado. After simulation results comparison for all the filter structures, it is shown that the best-optimized filter among all is FIR filter using WT multiplier.**

***Index Terms*- Adder; Multiplier; Filter; CSD (canonic sign digit); RCA(ripple carry adder); Xilinx.**

1. INTRODUCTION

With the explosive growth in mobile computing, portable multimedia applications and image & video processing the demand for compact and high-speed digital signal processing.(DSP) systems has been increased.[1]. Digital filters are mainly used for modification of signal attributes like noise and interference removal from the original signal. Digital filters are more accurate, highly stable and versatile therefore a preferred technique over its analogue counterpart.[2]. Optimization of the area, delay, and power of both digital and its analogeous building blocks become a challenging task with the increase in circuit components, one such application is designing FIR filter.[3, 4]. In any communication system, Finite impulse response.(FIR) filter is the basic building block and one of the most widely performed operations in DSP systems is filtering.[5]. After canonical sign digit.(CSD) representation of FIR filter coefficient many previous works.[6, 7] which focused on FIR filters design using various multipliers has been implemented like, a sharing multiplication approach.[8,9] which is a combination of add & shift operations over to the common computation results. However, sharing multiplication method has a major problem that is, as the the number of bits increase i.e. used to represent filter coefficients, for computation sharing we need an additional large memory area. Therefore, in the presented work, we prefer the multiplication of filter input with coefficients using add & shift method with canonical signed digit.(CSD) representation of filter coefficients.

The speed of FIR filter processor and execution time is maily depends on the speed of multiplier.[2]. Therefore, performance anlayses for different multiplication techniques are required in order to optimize filter area, delay and power. We have analyzed the performance of Vedic Mathematics used in ancient India for the multiplication of two decimal numbers'[10]. In previous works'[11], it was observed that the filter designed with VM and BK(Brent Kung) adder occupies more area but consumes lesser delay than the filter design with VM and RCA. Therefore for further optimization of conventional multiplication method, add and shift method is used instead of VM, because of its

efficiency and for this purpose barrel shifter adder is used [12]. Filter coefficients representation is done in more efficient CSD form instead of traditional binary representation [13]. Subsequently, RCA and parallel prefix adders are used for the addition of the delayed versions of those multiplications a. Finally, FIR filter is design using Booth and Wallace tree.(WT) multiplier as they are high speed multiplication method to multiply two numbers [14].

The rest of the paper is organized as follows: Section 2 gives the brief overview of FIR filter structure. In Sections 3–6, proposed FIR filters designed using different multipliers are explained with their components. Section 7 focused on simulation results and discussion of different filter structures. Finally the best of proposed work done is concluded in Section 8 followed by the references.

II. FIR FILTER

FIR filter is a digital filter with impulse response of finite duration and digital inputs as parameters. These filters also known as non-recursive filters because there is no feedback required.

Output of filter is defined as:

$Y[n] = X[n] * H[n]$ (1)

For M-order FIR filter, each value of the filter is represented as the weighted sum'of the most recent values of the input as:

$$Y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) = \sum_{k=0}^{M-1} x(k)h(n-k) \quad (2)$$

Here, $x(n)$ is the output. sequence, $h(n)$ represents the. coefficients of filter and $y(n)$ is the output .response [15]. FIR. filter can be designed in mainly .two structures: direct. form and transposed form. Direct form FIR. filter is area. efficient while the transposed form filter is delay efficient [16]. Transposed. form. FIR structure is shown in Fig. 1.

As shown in Figure 1 the input of .the. filter is multiplied. with the filter coefficients.at the same.time and then.these multiplications.are added together.to find the output sequence.of.the filter. To reduce area, delay.and the design.complexity of the filter, symmetric coefficients are taken. For Example, a.16-ordes FIR.filter with density.factor 20 is designed by.using the equiripple.method for the set of.frequencies given in Table 1.

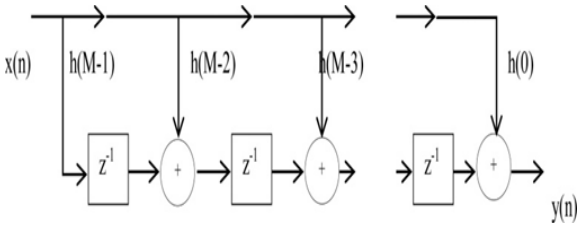

Fig 1. Transposed form FIR Filter Structure

For these.filters, both passband.and stopband.gain are equals to unity. When the.filter has linear.phase, the impulse.response $h(n)$ of FIR filter satisfies either.symmetric or asymmetric.condition given as:

$h(n) = \pm h(M - n - 1)$ (3)

For this.system, when M is.even, the no. of multiplications has been reduced.from M to $M/2$ .and for odd value of M, it has been reduced to $(M-1)/2$. Therefore, linear phase.symmetric structure further reduces.the area, delay and.power of the filter.

Table 1. FIR filter design parameters

Filter parameter		Frequency, Hz	
	I	II	III
sampling frequency	22,000	44,000	10,000
passband frequency	4000	10,000	2000
stopband frequency	4500	11,000	2500

The FIR. Filter that is designed. Using set-I frequencies. Listed in Table 1 has symmetric. Coefficients [42, 123, 71, 6, 77, 51, 105,298, 387, 298, 105, 51, 77, 6, 71, 123, 42] which reduces.area and power consumption. The coefficients. For filter with set-Infrequencies are [36, 130, 46, 55, 21, 104, 23, 318, 477, 318, 23,104, 21, 55, 46, 130, 36] and for set-III frequencies, the coefficients. Are [43, 35, 50, 30, 36, 96, 52, 309, 453, 309, 52, 96, 36, 30, 50, 35,43]. The filter order is M and length. Of the filter is $M+1$. The complexity of the filter. Increases with increased. Filter order because higher order filter consumes more time for signal processing. The magnitude response of FIR. Filters with I set of. Frequencies is shown in Fig. 2.

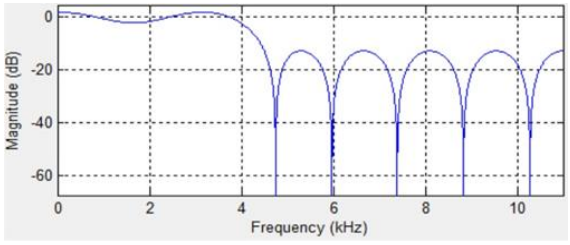

Fig. 2 Magnitude. Response of FIR. Filter with set-I frequencies as listed in Table 1

III. FIR"FILTER WITH ADD AND SHIFT.METHOD

Add & shift method is used'instead.of multiplication techniques to optimize area and delay. CSD representation of filter coefficients is used instead of binary representation because of its more efficiency. For shifting operation barrel shifter is used and for addition RCA and different parallel prefix adders are used for the delayed version of'these multiplication terms.

1.1CSD No. Representatio

A"radix-2 system with weights'{1, 0, −1} is known as CSD representation of a number. In CSD representation two consecutive non-zero bits cannot be there. There is min. no. of non-zero elements in CSD"representation that results in lesser no of adders. Therefore hardware implementation of Binary numbers is more efficient in CSD"representation'[13]. CSD"representation of the coefficients is done as:

$$X_{10} = \sum_{r=0}^{B-1} x_r 2^r \quad (4)$$

Where x_r = 0, 1, -1(denoted as $\bar{1}$).

Compared to binary representation, there are 33% lesser non-zero bits in CSD representation, means lesser hardware requirement'[17]. To represents a signed digit no. in CSD representation a property 'M' should be satisfied which states that product of any two consecutive bits should be zero.

1.2 Barrel shifters and adders

Barrel'shifter is used for shifting'and rotation of multiple bits'in a single clock cycle, as an integral component in some computing devices. In a single clock cycle, it can shift"any binary number or any data word either in clockwise direction or in anti-clockwise direction by any'specified number of bits'[18].

RCA is an adder circuit where for the addition of two *n*-bit numbers we used n full adders. If the input carry bit of the first full adder is zero i.e., $C_{in} = 0$, it can be'replaced by a half adder to reduce complexity. For addition of smaller bit numbers it is the simplest adder but for addition of large bit numbers it is not sufficient to use RCA because as.bit.length increases, propagation delay also.increases and this is the main'drawback of RCA'[19]. To"overcome the problem of"RCA, carry look ahead adder is used to generate Parallel Prefix adders which are more flexible and faster than RCA and these adders are used to reduce the'delay of the filter'[20]. The processing of parallel prefix adders is divided into three stages and some of these are explained as follows:

1. Initial stage is the Pre-processing stage for every parallel prefix adder [21]. In this stage 'propagate' signal"(P_i) and 'generate' signal (G_i) are generated"for every input bit pair"from the inputs by the given"equations:

$P_i = A_i \; xor \; B_i$	(5)
$G_i = A_i \; or \; B_i$	(6)

2. Carry generation network is the 2nd stage, it is used to generate the carry for each bit, carry generate"and carry propagate signals are generated for each"bit in each stage. For each stage, the carry propagate and carry"generate bits for every bit is"calculated in parallel [20, 21]. Carry generate (CG_i) and carry propagate (CP_i) signals are calculated as:

$$CP_i = A_i \; and \; A_{i-1} \quad (7)$$

$$CG_i = (A_i \; and \; B_{i-1}) \; or \; B_i \quad (8)$$

3. Post-processing'stage i.e. the'final"stage is used to calculate.the output of the adder. The output"carry for each bit is same as"the carry generate signal.of the same.bit in the last carry generation network stage as.defined in (9). The output bit is.calculated by the XOR.operation of propagate.signal of that bit and output carry of the previous.bit as given in the following equations

$$C_i = CG_i \quad (9)$$

$$S_i = P_i \; xor \; C_{i-1} \quad (10)$$

I. FIR Filter With Vedic"Multiplier

The word'Vedic.is'derived.from a Sanskrit.word 'Veda' means knowledge in Vedic.mathematics which is part of four Vedas and it is mainly based,on 16 principles'called 'sutras'"[22]. One of them is Urdhva.Triyakbhyam Sutra which we are using to design VM in this paper. In ancient India, UT sutra was used to multiply two decimal numbers, English meaning of this is 'vertically and clockwise'.[10, 23]. FIR filter using VM is designed in transposed form structure and, for providing delay, D Flip-flop is used whereas, for addition two different adders are used for delayed version of those multiplications. The

filter designed using VM and RCA consumes lesser area and power as compare to the filter with BK adder instead of RCA but delay is more compare to filter with BK adder. Therefore, we have designed two filters here, where the first filter components are VM, D flip-flop and RCA and the second filter is designed with BK adder instead of RCA.

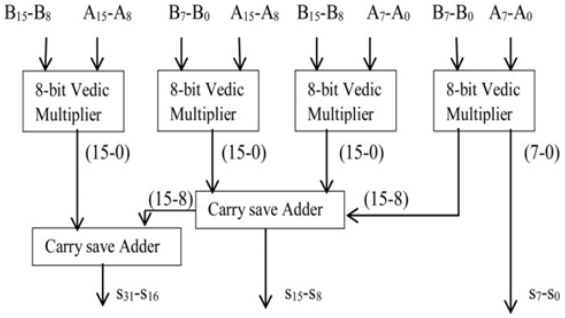

Fig. 3-16-bit Vedic Multiplier using CSA

Here we are designing a 16 order multiplier with the help of four *n/2* order multiplier i.e. 8 order multipliers. Initially both the inputs are divided in two sequences of upper and lower bits, then 8 order multipliers are"used as shown in Fig. 3 [24]. The outputs of 8-bit multipliers are added by using carry save adder in'a manner as shown in the figure 3. All 8-bit multipliers are also designed in the same way by using 4-bit multipliers and 8-bit RCAs. Similarly, 2-bit multipliers and 4-bit RCAs are required for the design of 4-bit multipliers and to design these 2-bit multipliers half'adders are used [10]. In 16-bit VM design we use two carry save adders where first CSA has three inputs and the second CSA has two inputs only. When addition of three or more numbers has to be done carry save adder'(CSA) is preferred as it provides faster addition than conventional adder or RCA [25]. The filter which is design using VM and RCA requires lesser area and power but more delay as compare to the filter with VM and BK adder.

IV. FIR FILTER WITH WT

In this FIR" filter design, Wallace Tree(WT) multiplier is used to reduce power and area using multiplication technique with good accuracy and less power consumption. Compressor is used here during multiplication process which is the basic building block used to accumulate the partial products. The multiplier architecture of WT is consists of the steps partial product generation, partial product reduction and partial product accumulation. In the partial product reduction stage the latency of the Wallace tree multiplier is reduced by decreasing the number of adders and after the multiplication, RCA is used to add'the delayed version of.these multiplications. D flip-flops are used as delay element. This filter requires more'area but lesser delay. Therefore this"filter has mainly three'components: WT, D flip-flop and RCA.

1.1 Wallace tree multiplier

WT is a high'speed"multiplication technique [14,26,27] in which multiplication of.two numbers is performed using half'adders and full adders by following these'three steps:

i. In this multiplication step each bit of one number is multiplied with each bit of the other, results n^2-bits if both numbers contain n-bits. Depending on the bit positions each wire carries a weight.

ii. Subsequently, the number of partial'products is reduced by using full adders and half adders. The three wires of same weight combined together and replaced with a full adder which provides an output of same weight and an output wire of higher weight. Half adder is used instead of Full adder if only two wires are available and this process is repeated until there are only two layers of partial products.

iii. Conventional adders are used to add these two layers.

Here a 16 order WT is designed to multiply the 16 bit input with filter coefficients which generates 32 bit output. It is observed that, for the range of frequencies this filter structure is faster and uses minimum power.

V. FIR FILTER WITH BOOTH MULTIPLIER

An algorithm which multiplies two signed binary numbers represented in 2's complement notation is known as Booth's multiplication technique. We are taking here an example of FIR filter structure with fixed filter coefficients where *L* coefficients of a given structure can be grouped into one MCMU block. Therefore, the original MCMU block is'split into *L* sub-MCMU"block, where each sub-MCMU block contains *N=L* coefficients. To'generate *L* products the"*L* sub-MCMU"blocks are summed up in

the form of product-accumulation“unit by an“adder tree“using (*L*-1) PSTAs.

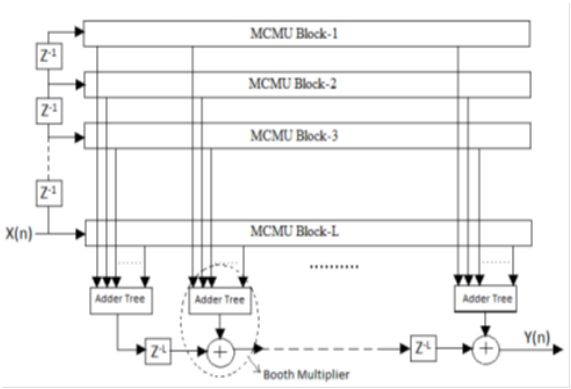

Fig. 4 FIR filter structure with MCMU blocks and booth multiplier

Figure 4 “shows“ the “structure of the“ implemented “filter whose output is generated by the accumulation of results of adder trees. In this structure“approximately about“a ratio“of (*L*-1)*/L* of“STAs in the TDF“structure are“replaced with PSTAs in that. More“replacement of'STAs with“PSTAs means“more“area“saving“since“the word-length of PSTAs is“generally smaller“than their“corresponding STAs. Note that the word-length of the PSTAs“increases with *L* attributed to’the extension“of the sizes of the“intermediate’results. In this paper, we’have done“implementation of a 16-tap“filter, the MCMU“block is being“divided into 16 sub-MCMU blocks and the partial products from”all the MCMU blocks are“accumulated in the product accumulation“unit.

The adders in the adder“trees“along with the“STAs can be implemented“using Shift-add“multipliers or“Booth Multiplier to“reduce“the“critical“path“delay. We have“implemented the filter structure”using both of“these and made“a comparison of the results with’the’conventional”method.

VI. SIMULATION RESULTS AND DISCUSSION

In this section we will show the simulation results for three different set’of’frequencies shown in Table1 of all the proposed filters discussed. The area, delay and power of all the FIR filters using different adders are compared in Table2. FIR filter is proposed for three set of frequencies using different adders and multipliers as shown above in sections 3-6. Filter coefficients used in multiplication are calculated in”MATLAB”FDA toolbox by designing a’low pass filter for given set of frequencies. The FIR filter design with these coefficients is designed’in Xilinx Spartan 3E XC3S500E’by using different multiplication’techniques mentioned’in Sections 3–6.

Table 2. Comparison of FIR Filter for different multipliers

Attribute	For Freq. Set	FIR with CSD & BS	FIR with VM & RCA	FIR with VM & BK	FIR with WT & RCA	FIR with Booth Mul.
No. of Slices	I	1331	1341	1589	1473	1924
	II	1218	1298	1515	1575	1965
	III	1211	1354	1517	1530	1911
No. of Flip – Flops	I	528	483	480	466	653
	II	515	499	498	481	623
	III	520	478	477	473	593
No. of 4 I/p LUTs	I	2543	2481	2989	2722	3069
	II	2343	2333	2845	2914	2908
	III	2321	2312	2824	2801	2804
Min. Period, ns	I	9.25	9.97	9.69	9.81	3.94
	II	9.18	11.41	9.75	8.01	3.69
	III	9.17	8.86	8.79	7.98	3.74
Max. Freq., MHz	I	107.95	100.21	101.26	101.23	253.16
	II	108.70	87.38	102.78	124.57	271.0
	III	108.85	112.85	113.79	124.64	267.37
Power, Watt	I	54.66	32.99	42.66	35.19	48.68
	II	51.79	50.36	59.69	54.78	60.49
	III	52.75	50.49	59.67	51.47	61.06

The area, delay and power of these filters’ with’ different methods are compared in Table2. Powers of ’all’ filters are calculated in Vivado- 2014.4. The energy delay product’(EDP) of BK adder based filter is approx. 72.16% higher to the EDP of WT multiplier ’based’ filter. WT multiplier based filter design has the best EDP and it’is ’a’ high speed multiplication method among all filters because it has minimum delay for the range of frequencies and it reduces one stage of multiplication bits by processing all full’ adders and half’ adders at the same time count. But it occupies slightly more area approx. 36.45% more than the filters designed with add & shift method. Filter designed with booth multiplier provides a reduction in delay of approx. 68.26% over the conventional method. Filter designed with BK adder and VM has approx. 11.61% less area than the filter with WT multiplier but it has approx. 43.89% more EDP than WT multiplier. However, the result of filter designs is slightly differing from the result of adder & multiplier designs, because of defined

coefficients. After the simulation results of all the filter structures it is observed that the performance of the filters can't be same but it varies with selected coefficients of the Filters.

VII. CONCLUSION

In this paper, FIR filter using different multipliers architecture with low time complexity has been implemented efficiently. Comparison based on area, delay and power of all the proposed filter architectures for different frequencies is done in detail. After analyses it is observed'that,WT multiplier is'best multiplication method based on the calculation of EDP of all. Although filter designed with VM and RCA for'set-I frequencies, has minimum EDP but filter designed with WT has better performance for some range of frequencies and also has minimum delay. Filter with Booth multiplier provides reduction in delay of approx. 68.26% and filter with BK adder also provides good result in terms of delay and fastest among all but EDP is approx. 43.89% more as compared to the WT based filter. FIR filter with VM & RCA consumes minimum area'but it provides maximum delay among all'adders and its EDP is approx. 72.16% higher'than that of WT multiplier based filter. Therefore after considering all the factors such as'area, delay and power, it is observed that FIR filter designed with WT multiplier & RCA has better performance'among all the proposed FIR filter structures.

REFERENCES

[1] Maskell, D.L.: 'Design of efficient multiplierless FIR filters', IET CircuitsDevice Syst., 2007, 1, (2), pp. 175–180.

[2] Proakis, J., Manolakis, D.: 'Digital signal processing' (2008, 4th edn.), Pearson Prentice Hall

[3] Nandi, A., Saxena, A.K., Dasgupta, S.: 'Enhancing low temperature analog performance of underlap FinFET at scaled gate lengths', IEEE Trans. Electron Devices, 2014, 61, (11), pp. 3619–3624.

[4] Nandi, A., Saxena, A.K., Dasgupta, S.: 'Analytical modeling of double gate MOSFET considering source/drain lateral Gaussian doping profile', IEEE Trans. Electron Devices, 2013, 60, (11), pp. 3705–3709.

[5] Nekoei, F., Kavian, Y.S.: 'Some schemes of realization digital FIR filters on FPGA for communication applications'. IEEE Crimean Conf. on Microwave and Telecommunication Tech., September 2010, pp. 616–619.

[6] Wu, L., Cui, Y., Huang, J.: 'Design and implementation of an optimized FIR filter for IF GPS signal simulator'. IEEE Conf. on Microelectronics and Electronics, September 2010, pp. 25–28.

[7] Lim, Y.C., Parker, S.R.: 'FIR filter design over a discrete power-of-two coefficient space', IEEE Trans. Acoust. Speech Signal Process., 1983, ASSP-31, pp. 583–591.

[8] Samueli, H.: 'An improved search algorithm for the design of multiplierless FIR filter with powers-of-two coefficients', IEEE Trans. Circuits Syst., 1989, 36, pp. 1044–1047.

[9] Park, J., Muhammad, K., Roy, K.: 'High-Performance FIR filter design based on sharing multiplication', IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2003, 11, (2), pp. 244–253.

[10] Poornima, M., Patil, S.K., Shridhar, K.P.: 'Implementation of multiplier using vedic algorithm', Int. J. Innov. Technol. Exploring Eng., 2013, 2, (6), pp. 219–223.

[11] Mittal, A., Nandi, A.: 'Design of 16-bit FIR filter using Vedic multiplier with carry save adder'. Proc. of 44th IRF Int. Conf., November 2015, pp. 57–60.

[12] Goel, N., Nandi, A.: 'Design of optimized FIR filter using FCSD representation', Int. J. Electr. Electron. Eng., 2015, 2, (1), pp. 3–6

[13] Chang, C.H., Faust, M.: 'A new common subexpression elimination algorithm for realizing low complexity higher order digital filters', IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 2010, 29, (5), pp. 844–848.

[14] Kumar, M.R., Rao, G.P.: 'Design and implementation of 32 bit high level Wallace tree multiplier', Int. J. Tech. Res. Appl., 2013, 1, (4), pp. 86–90.

[15] Hsiao, S.F., Jian, J.H.Z.: 'Low cost FIR filter designs based on faithfully rounded truncated multiple constant multiplications', IEEE Trans. Circuits Syst.-II Expression Briefs, 2013, 60, (5), pp. 287–291.

[16] Bharti, D., Gupta, K.N.: 'Efficient design of different forms of FIR filter'. Int. Conf. on Recent Trends in Information Technology, 2014
[17] Mahesh, R., Vinod, A.P.: 'A new common subexpression elimination algorithm for realizing low complexity higher order digital filters', IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 2008, 27, (2), pp. 217–229
[18] Das, S., Khatri, S.P.: 'A timing driven approach to synthesize fast barrel shifters', IEEE Trans. Circuits Syst., 2008, 55, (1), pp. 31–35
[19] Han, J.H., Park, I.C.: 'FIR filter synthesis considering multiple adder graphs for a coefficient', IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 2008, 27, (5), pp. 958–962
[20] Chakali, P., Patnala, M.K.: 'Design of high speed Kogge-stone based Carry Select adder', Int. J. Emerging Sci. Eng., 2013, 2, (4), pp. 34–37
[21] Kumari, P.C., Nagendra, R.: 'Design of 32-bit Parallel Prefix Adders', IOSR J. Electron. Commun. Eng., 2013, 6, (1), pp. '1–6
[22] Paramasivam, M.E., Sabeenian, R.S.: 'An efficient bit reduction binary multiplication algorithm using vedic methods'. IEEE 2nd Int. Advance Computing Conf., 2010, pp. 1–258.
[23] Anjana, R., Abishna, B., Harshita, M.S.: 'Implementation of Vedic multiplier using Kogge Stone adder'. Int. Conf. on Embedded Systems, 2014, pp. 28–31
[24] Mahalakshmi, R., Sasilatha, T.: 'A power efficient Carry Save adder and modified Carry Save adder using CMOS technology'. IEEE Int. Conf. on Computer Intelligence and Computing Research, 2013.
[25] Kumari, P.C., Nagendra, R.: 'Design of 32-bit Parallel Prefix Adders', IOSRJ. Electron. Commun. Eng., 2013, 6, (1), pp. 1–6
[26] Satish, C., Arur, P.C., Kumar, G.K.: 'An efficient high speed Wallace tree multiplier', Int. J. Emerging Trends Electr. Electron., 2014, 10, (4), pp. 38–42
[27] Sankar, D.R., Ali, S.A.: 'Design of Wallace tree multiplier by Sklanskyadder', Int. J. Eng. Res. Appl., 2013, 3, (1), pp. 1036–1040.
[28] Elisardo Antelo, Paolo Montuschi, IEEE, and Alberto Nannarelli, IEEE Improved 64-bit Radix-16 Booth Multiplier Based on Partial Product Array Height Reduction, 2017.