Asian Journal of Information Technology 15 (10): 1559-1569, 2016
ISSN: 1682-3915

Implementation of Discrete Wavelet Transform in SoC Using Vedic Mathematics

[1]R. Nirmala and [2]K. Sathiya Sekar
[1]Department of ECE, Vivekanandha college of Engineering for Women,
Ellayampalayam, Tiruchengode,
[2]Department of EEE, S A Engineering College, Chennai, Tamil Nadu, India

Abstract: A new design for the implementation of discrete wavelet transform in system on chip(SoC) using vedic mathematics has been proposed. The proposed design incorporates the efficiency of Vedic mathematics and low power consumption through 90nm technology in the SoC design. The study presents a design methodology for a custom ASIC Processor core for signal and image processing application with highest performance and lowest part cost. The system on chip is designed in 90 nm technology and consists of a MAC unit, SRAM and buffers The design mainly focused on the processor core block which is implemented using a MAC with a vedic multiplier, adder and shift register. The processor core is designed to perform the various filtering operation for the DWT. Buffers are placed in the data path to speed up operation and to provide high driving capacity to adjacent stages minimizing the loading effect. The performance of the various blocks of the discrete wavelet transform SOC is compared with other Processing elements and systems. From the results it is been shown that the proposed design is efficient when compared to the other methods.

Key words: System on chip, vedic multiplier, MAC, adder, vuffer, DWT

INTRODUCTION

In the field of signal and image processing throughout the year many research have been carried out for the implementation of SoC. Even though time analysis and frequency analysis are advantages in their own aspects the wavelet transform outperforms the efficiency of the former. The discrete wavelet transform is an efficient platform for multiresolution analysis with excellent characteristics in the time and frequency domain. When compared to Fourier analysis and discrete cosine transform, discrete wavelet transform has better coding efficiency and excellent quality of restoration of signal and image with high compression ratio and better noise removal characteristics.. The objective of this work is to design a SoC to perform 1D discrete wavelet transform operation.

Many works have been carried out for the design of SoC for wavelet transform. One such design is discussed in the study Marud and McCanny (2001) which presents the design of a processor for biorthogonal wavelet transform. In this research the core is designed based on lifting method, discrete lattice structure shift invariant method for wavelet filter. Other works focus on a design of optimized MAC unit proposed by Deepak and Kailath (2012) which reduces the propagation delay and power consumption. The MAC unit is designed such that the number of partial products reduces by 25%. A high performance MAC unit is proposed by Jagadeesh *et al*. (2013). In this study, a multiplier is designed using AND and XOR gates and compared with array and booth multiplier. A modified Wallace tree block is implemented in the MAC unit (Jagadeesh *et al*., 2013). A proposed MAC unit (Jagadeesh *et al*., 2013) is compared with other methods and found to be efficient. In another work (Veeramachaneni and Srinivas, 2013) ALU is designed using a new counter architecture multiplier which consumes less power and delay compared to existing multipliers. In another research (Srivastava *et al*., 2013) two different type of array multipliers are designed to reduce the power consumption and delay. The first multiplier is designed using a regular hybrid full adder and has better performance when compared to conventional array multiplier. The second multiplier is designed using new hybrid adder.

High speed Vedic multiplier design using Vedic mathematics is proposed by YogitaBansal. A multiplier designed for digital design using Urdhva Tiryakbhyam sutra will be efficient in power, area and speed. In the design of area efficient low power multiplier (Murugeswari and Mohideen, 2014) by modifying the carry save adder

Corresponding Author: R. Nirmala, Assistant Professor, Department of ECE, Vivekanandha college of Engineering for Women, Ellayampalayam, Tiruchengode, Tamil Nadu, India

circuit of a Wallace tree multiplier, the reduction in area and low power has been achieved. Likewise a truncated multiplier is designed with MUX based full adder instead of normal CMOS based full adder. A MAC unit designed with Baugh-wooley algorithm based multiplier and Braun based multiplier are discussed by Basiri and Sk (2014) and Warrier *et al.* (2014). In proposed structure, the conventional carry select adder is modified by using a BEC with RCA, for achieving better power consumption and area reduction (Parmar and Singh, 2013). In performance analysis and comparison of RCA based carry save adder and CLA based carry save adder is discussed (Javali *et al*., 2014).

In this research, a design for implementation of discrete wavelet transform SoC is presented. The design architecture consists of a wavelet transform processor (which includes a multiplier, adder, coefficient selecting multiplexer and shift registers), memory, switch and buffer, multiplexer and control unit. The processor unit is designed using multiplier like array, wallace tree, Baugh-wooley, Braun and Vedic multiplier. The adder structures are ripple carry, carry save, carry select, carry skip and carry look-ahead adder. The memory cell is designed in 90nm CMOS technology. The study is organized in this manner. The second subdivision presents the basics of DWT and its blocks. The third subdivision discusses the theory and implementation of Vedic mathematics. The fourth subdivision presents the architecture of SoC and DWT processor followed by the proposed architecture. Research findings are presented in the results chapter followed by the conclusion and reference.

Discrete wavelet transform: In signal image processing field noise removal and extraction of features are one of the most needed processing methods for efficient data analysis. The hardware implementation of 1D wavelet transform is been done using convolution based and lifting based architectures. For 2D DWT implementation direct and line based architectures are adopted. In most of the work lifting scheme is followed for the implementation of 2D DWT architecture in VLSI technology. But, even though lifting scheme out performs convolution scheme. The critical path is longer than the convolution based scheme. The overloading architecture increases if pipelining is introduced to reduce the critical path. Since, pipelining requires large number of registers.

In order to achieve a critical path with lesser number of multiplier, at least four pipelining stages are required for one lifting step which is area consuming and in addition to that a large temporal buffer is needed. The few options left behind is the modification in the lifting scheme through recombining of intermediate results which is complex. For 2D-DWT architecture the complexity is still increases, the limitations in the number of temporal buffers with larger size to achieve the intermediate results is one disadvantage of lifting scheme. The flipping structure also inserts one multiplier delay for every pipelining. In addition minimizing the number of multiply operations reduces the execution time only in program implementation and not in VLSI implementation.

The analysis of wavelet architecture: The 2 dimensional DWT is implemented using 1D-DWT twice on the row vice data and column vice data. VLSI architectures for implementing the filter banks consists of low pass filter, high pass filter, decimators, expanders and delay elements. Efficient design should have lesser area, low power consumption and high throughput.

Filter bank implementation: A filter bank is implemented with various methods like direct form structure, polyphase decomposition, lattice structure etc. The direct form structure is inefficient due to the number of multiplication required when compare to other architecture. In addition to that due to decimation by two computation of unused N/2 samples become redundant. The disadvantages are rectified by the poly-phase structure design where the filter coefficients are split into odd and even samples. The split samples are convolved with one another and the two phases finally summed to get the filter output. The poly-phase structure reduces the number of computation by half and the efficiency is increased by 50%. The poly-phase structure design is considered in this researc for the implementation of DWT architecture. The work is extended to increase the efficiency and throughput by the use of lattice structure which is efficient when compared to the poly-phase structure. The number of multiplications and coefficients are reduced in this structure. Figure 1 gives the basic structure of direct form 3 tap-broadcast FIR filter .The architecture can be further improved by applying refining and pipelining. In fine grain pipelining structure (Fig. 2) the multiplier is broken into smaller units. The polyphase decomposition technique is addressed for the implementation of the DWT as the method is well suited for the multirate signal processing. An N-tap FIR filter can be expressed as (Fig. 3):

$$Y(z) = H(Z)X(Z) = \sum_{n=0}^{N-1} h(n)z^{-n} \sum_{n=0}^{\infty} x(n)z^{-n} \qquad (1)$$

The input sequence {x(0), x(1), x(2), x(3),…..} can be decomposed into even numbered and odd numbered part as follows:

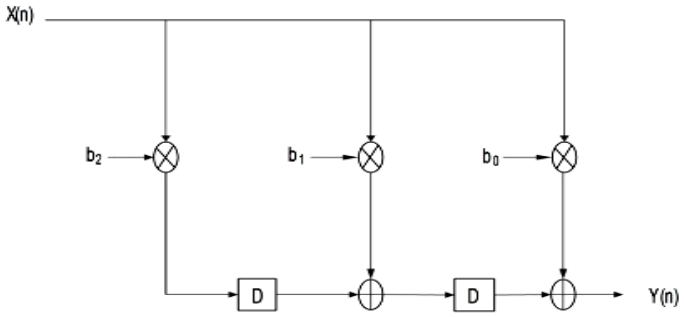

Fig. :1 A direct form structure for 3 tap filter

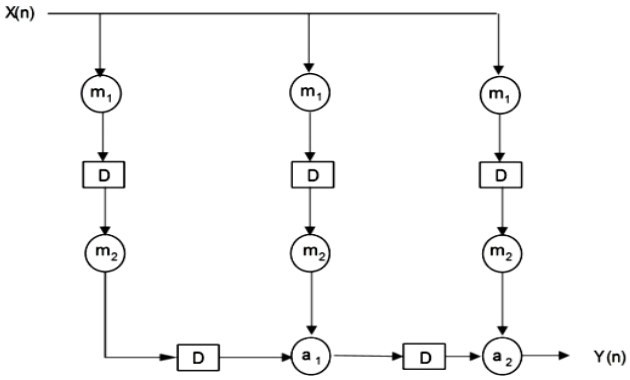

Fig. 2: A fine grain pipelining structure for 3 tap filter

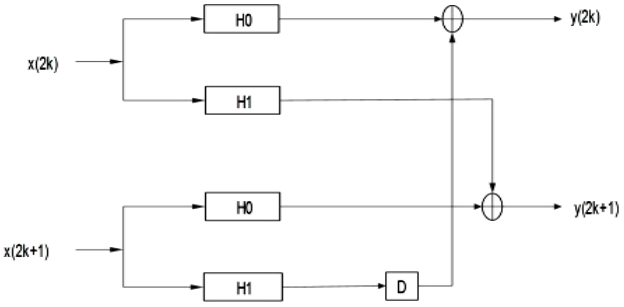

Fig. 3: Polyphase decomposition technique

$$X(Z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \ldots$$
$$= X_0(Z^2) + Z^{-1}X_1(Z^2) \quad (2)$$

X(Z) is decomposed into two polyphase. The length of N filter co-efficient H(Z) is decomposed as:

$$H(Z) = H_0(Z^2) + Z^{-1}H_1(Z^2) \quad (3)$$

Where, $H_0(Z^2)$ and $H_1(Z^2)$ are of length N/2 and are referred to as even sub filter and odd sub filter respectively. These two functions from the base for the filter banks designed using polyphase filter. The output sequence is computed as :

$$Y_0(Z^2) = X_0(z^2)H_0(Z^2) + Z^{-2}X_1(z^2)$$
$$Y_1(Z^2) = X_0(Z^2)H_1(Z^2) + X_1(Z^2)H_0(Z^2) \quad (4)$$

The polyphase decomposition can be used to derive L-parallel FIR filters by decomposing the inputs, impulse response and output into the L subsequences as:

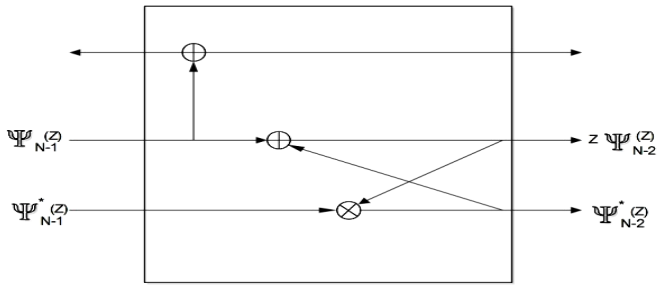

Fig. 4: The lattice structure of an Nth order basic filter

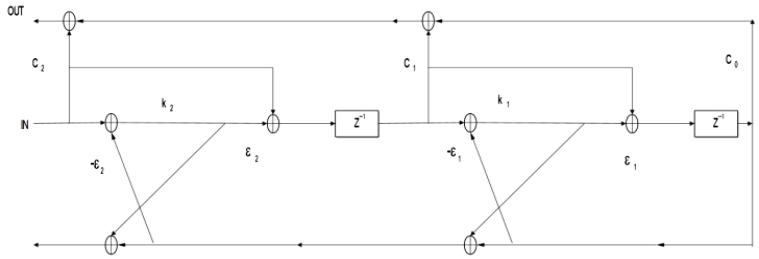

Fig. 5: second order lattice multiplier

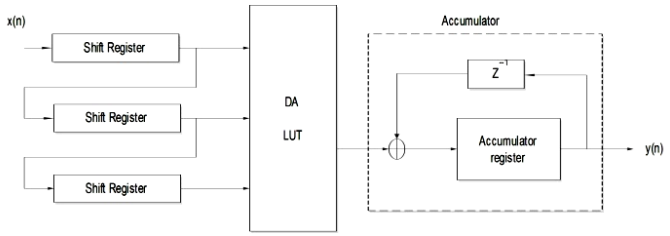

Fig. 6: Distributed architecture

$$X_i(Z)=\sum_{k=0}^{\infty} Z^{-k}x(Lk+i), i=0,1,2...L-1 \quad (5)$$

$$H_j(Z)=\sum_{k=0}^{\frac{N}{L}-1} Z^{-k}h(Lk+j), j=0,1,2...L-1 \quad (6)$$

$$Y_i(Z)=\sum_{k=0}^{\infty} Z^{-k}y(Lk+i), i=0,1,2...L-1 \quad (7)$$

The L output sequence is implemented using a combination of L sub filter banks from the L inputs subsequences:

$$Y_k = Z^{-L}\sum_{i=k+1}^{L-1} H_i X_{L+k} + \sum_{i=0}^{k} H_i X_{K-j}, 0 \le k \le L-2 \quad (8)$$

The L-parallel FIR filter requires L^2 sub filtering operations. The length is N/L and requires N/L multiplication and addition processing elements, so the L-parallel FIR filter requires $L^2N/L=LN$ multiply add operation. Here, the multiplication is reduced at the expense of increasing the number of additions required. The lattice structure of an N-th order basic filter is given in Fig. 4. A second order lattice multiplier is shown in Fig. 5.

The multiplier adder is placed in the feed forward section which can be pipelined by placing n-1 latches at the appropriate feed forward cutest location. The multiplier is further divided to reduce the computation time. The architecture for filter bank is so far discussed uses the multiplier and adder unit. In this research the filter architecture is designed using vedic mathematics In non-adaptive systems where filter coefficients are known in priori. The distributed arithmetic is well suited. In the architecture the inner sum of products is rearranged so that the multiply and accumulate operation is replaced with look up table memory calls, two's complement shift and add blocks. For example architecture is given below which a modified distributed architecture (Fig. 6) to perform computation with high speed is called DA

architecture. Resources are been shared among logic computations. Even though the area occupied by the multiplier is replaced by LUT memory, the memory size increases based on the filter sample rate and filter taps. The lookup table method consumes more area, since the provision to be done to store the sums of the products of N-bit input vectors and all possible filter coefficient combinations. Unlike conventional MAC type filter the throughput depends on the input bit precision and not on the filter length. So even when the filter length increases the throughput remains the same, while the logic resources increases. The parallel distributed architecture is given by:

$$Y = \sum_{n=0}^{N-1} C(n)x(n)$$
$$Y = \sum_{n=0}^{M-1} 2^{M} \sum_{n=0}^{N-1} C_{m}(n)x(n) \quad (9)$$

The poly phase form of the filter can be obtained by splitting the filters and input x (n) into even and odd phases. The filter bank architecture so far discussed can be improvised using parallel structure and pipelining through which computation complexity can be reduced going speed.

MATERIALS AND METHODS

Processing element for filter bank: In convolution and lifting based scheme the filter bank is implemented using multipliers, adders and shift registers. Single processing element architecture for a direct form filter is given in Fig. 7 for VLSI implementation.

Adder architecture: An N-bit ripple carry adder is implemented using N full adders where N is the total number of bits and the signal propagates from LSB to MSB. The carry traverses longest path known as worst case delay path via N-stages. The propagate time is calculated as:

$$t_{adder} = N - 1t_{carry} + t_{sum} \quad (10)$$

Where, t_{carry}- time taken to propagate carry, t_{sum}- time taken to generate the sum. To avoid carry rippling effect and to improve the speed of the parallel adder, Carry Look ahead adder is used. Carry for each cell increases the complexity as the number of bits increases and thus area is increased. Even though the trade-off's power, latency and area exist CLA is one of the fastest adder structures. To perform arithmetic functions faster in data processing processor Carry Select Adder (CSA) is used. It is a combination of RCA and MUX logic.

Multiplier architecture: All convolution based architecture require MAC unit while the distributed and lifting scheme uses adder and shifter processing elements to perform DWT. In this research a DWT architecture is proposed which uses Vedic multiplication. Multipliers are classified into ''shift and add'' multipliers, parallel and array multipliers. The new proposed Vedic multiplier is implemented for the DWT SOC design. Various structures and algorithms were so far formulated for performing multiplication operation. Array multiplier is very familiar because of its regular structure. Array multiplier structure is working based on repeated addition and shifting principle. The multiplication of each multiplier digit with multiplicand generates a partial product. Those partial products are added, before it is shifted into their bit sequence. A normal carry propagation adder is used to perform the summation. In array multiplier, N is the

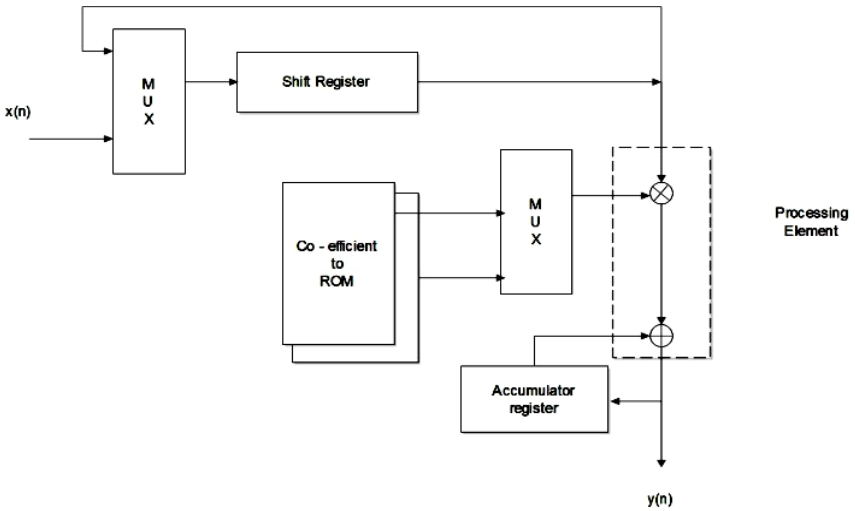

Fig. 7: A single proce

number of multiplier bits and N-1 adders are required for implementation. To design a regular multiplier Baugh wooley algorithm has been developed and it's suited for 2's complement number.

Braun multiplier is the simplest parallel multiplier. Cascaded carry save adders collect the computed partial products and these partial products are computed in parallel. Propagation time and depth of the carry save adder limits the completion time. Braun multiplier is only suited for positive operands. The rearranged tree structure of partial-sum adder reduced the requirement of number of adder cells and critical path. For larger multiplier the tree like fashion saves the substantial hardware realization and reduces the propagation delay. The propagation delay of the tree structure is equal to O (log 3/2 (N)). For larger multiplier word length it's substantially faster compared with carry save adder.

Vedic mathematics: The need of low power and high speed multiplier is increasing as the need of high speed processors are increasing. A conventional processor requires substantially more hardware resources and processing time in the multiplication operation, rather than addition and subtraction. This work presents a high speed Vedic Multiplier (VM) using sixteen sutras in Vedic mathematics ("UrdhvaTiryakbhyam"). A large number of high speed vedic multipliers have been proposed with urdhva tiryakbhyam sutra.

The Vedic multiplication technique is based on 16 vedic sutras or aphorisms, which are actually word formulae describing natural ways of solving a whole range of mathematical problems. The mathematical operations using, Vedic Method are very fast and requires less hardware, this can be used to improve the computational speed of processors. This paper describes the design and implementation of N x N bit Vedic multiplier based on Urdhva-Tiryakbhyam sutra (vertically and crosswise technique) of vedic mathematics using spice tool.

Vedic mathematics sutras: Vedic mathematics deals with sutras based on the mathematical operations. These sutras and their functions are detailed as anurupye shunyamanyat. If one is in ratio, the other is zero, Chalana-Kalanabyham differences and similarities, ekadhikina purvena. By one more than the previous one, ekanyunena purvena. By one less than the previous one, gunakasamuchyah. The factor of the sum is equal to the sum of the factors, gunitasamuchyah. The product of the sum is equal to the sum of the product, Nikhilam Navatashcaramam Dashatah. All from 9 and the last from 10, Paraavartya Yojayet. Transpose and adjust, Puranapuranabyham. By the completion or noncompletion, Sankalana-vyavakalanabhyam. By addition and by subtraction, Shesanyankena Charamena The remainders by the last digit, Shunyam Saamyasamuccaye. When the sum is the same that sum is zero, Sopaantyadvayamantyam. The ultimate and twice the penultimate, urdhva tiryakbyham vertically and crosswise, vyashtisamanstih part and whole, yaavadunam whatever the extent to fits deficiency. Among the above the urdhvatiryabhyam is the sutra which deals with the multiplication operation.

Urdhvatiryakbhyam-multiplication: Urdhvatiryakbhyam is a sanskrit word which means vertically and crosswire in English. The method is a general multiplication formula applicable to all cases of multiplication. It is based on a novel concept through which all partial products are generated concurrently. This sutra is based on "Vertically and crosswise" technique and makes almost all the numeric computations faster and easier. The advantage of multiplier based on this sutra over the others is that with the increase in number of bits, area and delay increase at a smaller rate in comparison to others. Due to its regular structure, it can be easily layout in a silicon chip and also consumes optimum area. As the number of input bits increase, gate delay and area increase very slowly as compared to other multipliers. Therefore, Urdhva Tiryakbhyam multiplier is time, space and power efficient. In Fig. 8, this method is illustrated with the multiplication of two numbers. The numbers of steps in the process depend upon the number of the digits being used. This type of multiplier is independent of the clock frequency of the processor because the partial products and their sums are calculated in parallel. The net advantage is that it reduces the need of microprocessors to operate at increasingly higher clock frequencies. As the operating frequency of a processor increases the number of switching instances also increases. This results in more power consumption and also dissipation in the form of heat which results in higher device operating temperatures. Another advantage of Urdhva Tiryakbhyam multiplier is its scalability. The processing power can easily be increased by increasing the input and output data bus widths since it has a regular structure.

The line diagram in Fig.8 illustrates the algorithm for multiplying two 4-bit binary numbers a_3 a_2 a_1 a_0 and b_3 b_2 b_1 b_0. The procedure is divided into 7 steps and each step generates partial products. Initially as shown in step 1 of Fig. 8, the Least Significant Bit (LSB) of the multiplier is multiplied with least significant bit of the multiplicand (vertical multiplication). This result forms the LSB of the

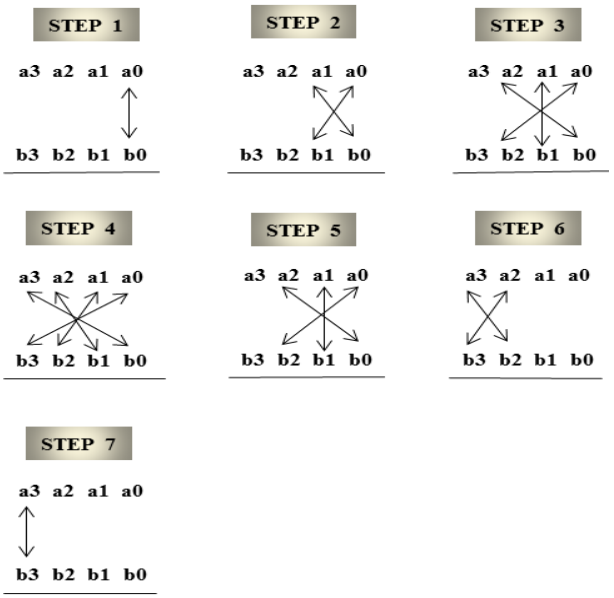

Fig. 8: Multiplication of two 4 bit numbers using urdhvaTiryak Bhyam method

product. In step 2 next higher bit of the multiplier is multiplied with the LSB of the multiplicand and the LSB of the multiplier is multiplied with the next higher bit of the multiplicand (crosswire multiplication). These two partial products are added and the LSB of the sum is the next higher bit of the final product and the remaining bits are carried to the next step. For example, if in some intermediate step, we get the result as 1101, then 1 will act as the result bit (referred as rn) and 110 as the carry (referred as cn). Therefore cn may be a multi-bit number. Similarly other steps are carried out as indicated by the line diagram. The important feature is that all the partial products and their sums for every step can be calculated in parallel. Thus every step in fig. has a corresponding expression as follows:

$$r0 = a0b0 \tag{11}$$

$$c1r1 = a1b0 + a0b1 \tag{12}$$

$$c2r2 = c1 + a2b0 + a1b1 + a0b2 \tag{13}$$

$$c3r3 = c2 + a3b0 + a2b1 + a1b2 + a0b3 \tag{14}$$

$$c4r4 = c3 + a3b1 + a2b2 + a1b3 \tag{15}$$

$$c5r5 = c4 + a3b2 + a2b3 \tag{16}$$

$$c6r6 = c5 + a3b3 \tag{17}$$

With c6 r6 r5 r4 r3 r2 r1 r0 being the final product

After this, only the least significant digit of the resulting number is taken as product digit and rest are considered as carry digits. Initial carry is taken as zero. Another technique for the calculation of Urdhva Tiryakbhyam method is shown in Fig. 9. In this technique, the numbers to be multiplied let us say 5498 and 2314 are written on the consecutive sides of the square table. On partitioning the square into rows and columns, each row/ column belongs to one of the digit of the two numbers to be multiplied such that every digit of one number has a small square common to the digit of other number. These small squares are further divided into two equal parts by crosswise lines. Now the each digit of one number is multiplied with every digit of second number and two digit products are placed in their corresponding square. The digits on crosswise line are added with previous carry. Digits on dotted significant digit of the resulting number are taken as product digit and rest are considered as carry digits. Initial carry is assumed to be ze ro here also .The Proposed VLSI Implementation is shown in Fig. 10.

For NXN multiplication unit, we require four N/2 bit multipliers, two N bit full adders, one half adder and N/2 bit full adder to add the sum and carry of half adder shown in Fig. 9. High speed of multiplier depends highly upon speed of adder units used.

Proposed SoC for DWT architecture: A new design for the implementation of discrete wavelet transform in system on chip using Vedic mathematics has been

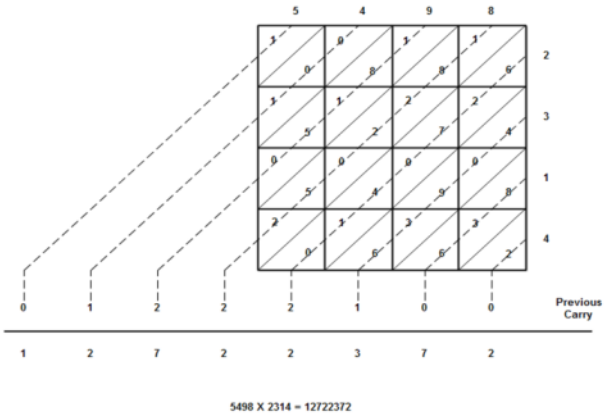

Fig. 9: Alternative way to calculate the Urdhva Tiryak Bhyam

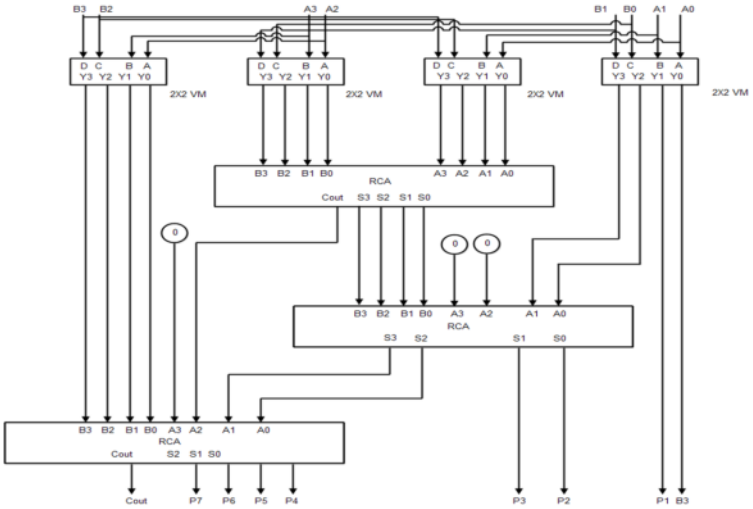

Fig. 10 . VLSI Implementation of the proposed vedic multiplication UrdhvaTiryakbhyam

designed. The proposed design incorporates the efficiency of Vedic mathematics and the reduced power consumption using 90nm. The design architecture consists of a wavelet transform processor (which includes a multiplier, adder, coefficient selecting multiplexer and shift registers), memory, switch and buffer, multiplexer and control unit. The processor unit is designed using vedic multiplier .The efficiency of the vedic multiplier is compared with multipliers like array, Wallace tree, Baugh-wooley, Braun and Vedic multiplier. Even though there is no much variation in power consumption of vedic multiplier with other multipliers the ease is computing is the main interest of the findings. As the vedic multiplier becomes the part of the MAC unit the rest of the blocks like adder structures are implemented using ripple carry, carry save, carry select, carry skip and carry look-ahead adder. The performance of the multiplier with the adder is promising. As the computation in DWT deals with filter banks there is plenty of requirements for the use of memory cell. Since, the occupation of memory block inside the chip occupies more area and cost increases the memory unit is designed for the storage of the co-efficient only. The memory cell is designed in 90nm CMOS technology. The processor is designed to perform the filtering operation of the DWT. The SRAM block is designed to operate at low power and high speed. Buffers are placed in the data path to speed up operation and to provide high driving capacity to adjacent stages minimizing the loading effect. The performance of the various blocks of the discrete wavelet transform SOC is compared with other systems. From the results it is been shown that the proposed design is efficient when compared to the other methods. The block diagram of the proposed SOC architecture of the DWT is given in the

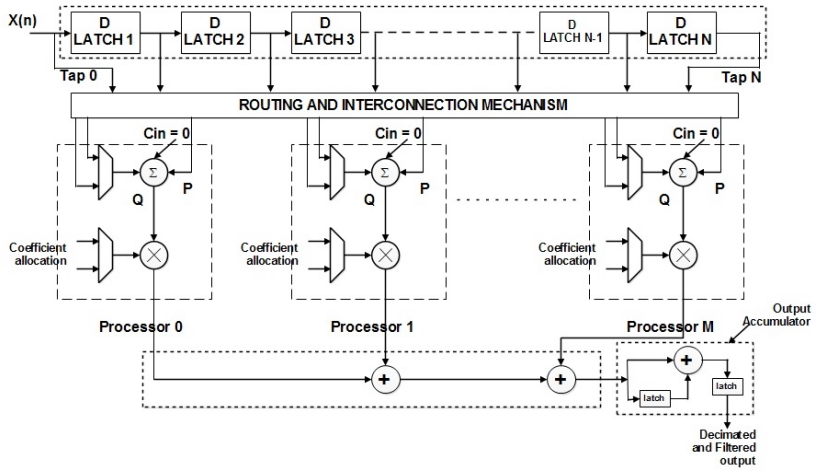

Fig 11. block diagram of the proposed SOC architecture of the DWT

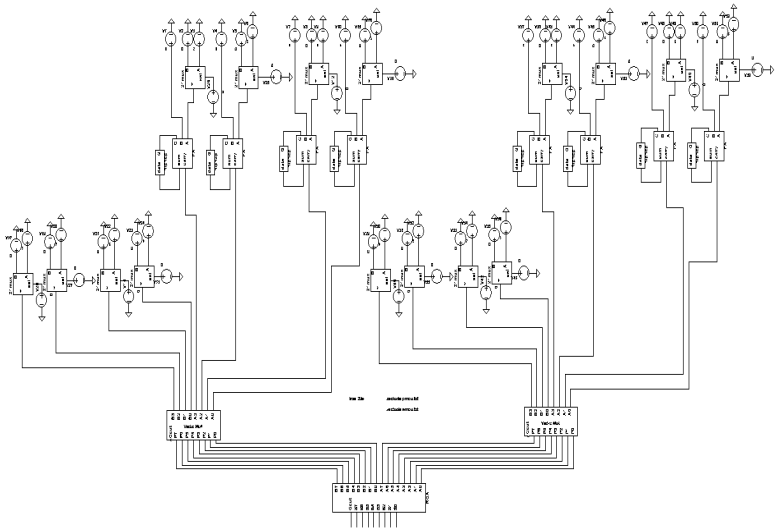

Fig. 12: Circuit diagram of the proposed processor core SOC architecture for the DWT

Fig. 11. The filter bank is generalized and is implanted to do the functionality of any type of filter architecture. The coefficient allocation/selection block is a multiplexer which choose the various operation to be performed like decimation, interpolation, low pass and high pass filtering. The summation block in the processor core is implemented using full adder while the summation block of the final output of processor M is implemented using Ripple carry adder. The individual block details of the processor core with respect to number of transistors and power consumed. The processor block forms the one stage of the decomposition tree of the discrete wavelet transform. The design for implementation for the SOC block is done to accompany maximum of 7 decomposition stages. For the size to be minimizes the memory block is external to the architecture.

RESULTS AND DISCUSSION

The proposed SOC architecture for the DWT is given in Fig. 11 and its circuit is given in the Fig. 12. The SOC is designed to consume less power and high speed .Even though vedic multiplier occupies more area in terms of number of transistors. The power consumed is less and execution speed is high when compared to other multiplier. The performance analysis for a 4x4 MAC unit design is compared in the Table 1 and 2. Table 1 displays the performance analysis of various multiplier used in the processor core of SOC. In Table 2 the performance analysis of 4x4 bit adder block is given. The RCA is efficient for vedic multiplication and consumes less

Table 1: Performance analysis of various multiplier in processor core (V_{DD} = 1V,Technology = 90nm)

Adder	No of transistor	Power dissipation in (μw)	Execution time (ns)
RCA	232	0.389	12.9358
CLA	316	28.575	17.2381
CSKIPADD	262	0.389	13.2287
CSAVEADD	274	0.288	12.7608
CSELADD	566	19.612	15.7092

Table 2: Performance analysis of various adder in processor core (V_{DD} = 1V,Technology = 90nm)

Multiplier	No of transistor	Power dissipation in (μw)	Execution time (ns)
Array	640	7.054	18.4865
Wallace	640	7.025	19.1858
Bough	666	10.23	18.9397
Braun	640	5.529	19.9811
Vedic	952	1.327	18.4436

Table 3: Block parameters and performance analysis of cores (V_{DD} = 1V,Technology = 90nm)

Processor Core System blocks	Number of transistors		Total power in (nw)	
	Processor0	Processor 1	Processor0	Processor 1
2:1 MUX	160	160	288.168	288.168
Full adder	232	232	115.3944	115.3944
Flip-flop	72	72	0.000346622	0.000346622
Vedic multiplier	952	952	259.648	259.648
Ripple carry adder	464		374.839	
Total	3296		1326.41	

powe. The performance analysis and parameter of the dual processor core SoC is given in the Table 3. As the processor follows "replica" design of DSP architecture, the power consumed by the cores remain same. From the above analysis it is been shown that the SOC designed for the DWT is new method and is efficient. All the designs are implemented and tested using LTspice tool.

CONCLUSION

The new processor unit design proposed here for the DWT SOC has significant advantages with regards to Power Consumed and speed over other conventional processor cores which used multipliers including booth recoded multipliers and array multipliers. Also the proposed design is found to be new methodology in terms of the algorithm used the Vedic mathematics. More studies on it can be done to further improve the efficiency of the unit. Future designs are to be done in design of mixed signal SoC which includes analog to digital converter, the proposed Vedic mathematics based MAC architecture and digital to analog converter.

REFERENCES

Basiri, M. and N.M. Sk, 2014. An efficient hardware based MAC design in digital filters with complex numbers. Proceedings of the 2014 International Conference on Signal Processing and Integrated Networks (SPIN), February 20-21, 2014, IEEE, Noida, India, ISBN: 978-1-4799-2865-1, pp: 475-480.

Deepak, S. and B.J. Kailath, 2012. Optimized MAC unit design. Proceedings of the 2012 IEEE International Conference on Electron Devices and Solid State Circuit (EDSSC), December 3-5, 2012, IEEE, Bangkok, Thailand, ISBN: 978-1-4673-5694-7, pp: 1-4.

Jagadeesh, P., S. Ravi and K.H. Mallikarjun, 2013. Design of high performance 64 bit mac unit. Proceedings of the 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT), March 20-21, 2013, IEEE, Nagercoil, India, ISBN: 978-1-4673-4921-5, pp: 782-786.

Javali, R.A., R.J. Nayak, A.M. Mhetar and M.C. Lakkannavar, 2014. Design of high speed carry save adder using carry lookahead adder. Proceedings of the 2014 International Conference on Circuits, Communication, Control and Computing (I4C), ovember 21-22, 2014, IEEE, Bangalore, India, ISBN: 978-1-4799-6545-8, pp: 33-36.

Masud, S. and J.V. McCanny, 2001. Design of silicon IP cores for biorthogonal wavelet transforms.. VLSI. Signal Process. Syst. Signal Image Video Technol., 29: 179-196.

Murugeswari, S. and S.K. Mohideen, 2014. Design of area efficient and low power multipliers using multiplexer based full adder. Proceedings of the 2014 2nd International Conference on Current Trends in E ngineering and Technology (ICCTET), July 8, 2014, IEEE, Coimbatore, India, ISBN: 978-1-4799-7986-8, pp: 388-392.

Parmar, S. and K.P. Singh, 2013. Design of high speed hybrid carry select adder. Proceedings of the 2013 IEEE 3rd International Advance Computing Conference (IACC), February 22-23, 2013, IEEE, Ghaziabad, Uttar Pradesh, India, ISBN: 978-1-4673-4527-9, pp: 1656-1663.

Srivastava, P., V. Vishant, R.K. Singh and R.K. Nagaria, 2013. Design and implementation of high performance array multipliers for digital circuits. Proceedings of the 2013 Students Conference on Engineering and Systems (SCES), April 12-14, 2013, IEEE, Allahabad, India, ISBN: 978-1-4673-5628-2, pp: 1-5.

Veeramachaneni, S. and M.B. Srinivas, 2013. Design of optimized arithmetic circuits for multiplier realization. Proceedings of the 2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), December 19-21, 2013, IEEE, Visakhapatnam, India, ISBN: 978-1-4799-2750-0, pp: 219-224.

Warrier, R., C.H. Vun and W. Zhang, 2014. A low-power pipelined MAC architecture using baugh-wooley based multiplier. Proceedings of the 2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE), October 7-10, 2014, IEEE, Tokyo, Japan, pp: 505-506.