VLSI IMPLEMENTATION OF LOW POWER HIGH SPEED DWT FIR FILTER ARCHITECTURE BASED ON DISTRIBUTIVE ARITHMETIC ALGORITHM

Manasa M G[1], Gayathri S[2], Aravind R[3], Shalini M G[4]

Article History: Received: 29.02.2023 **Revised:** 13.04.2023 **Accepted:** 06.06.2023

Abstract

Discrete Wavelet Transform (DWT) is wide used in signal and image processing applications. VLSI implementation of DWT architecture optimizing area, power and speed performances is carried out by design of customized architectures. In this paper, Distributive Arithmetic (DA) based algorithm is developed and a suitable architecture is designed considering symmetric property of filter coefficients. The improved method for DWT filter bank implementation occupies less than 1% of the total area required by direct implementation. The architecture is optimized for ASIC implementation and power dissipation is optimized. Power saving of 62% is achieved in implementing the design targeting 45nm technology. With low power filter bank design, DWT based signal and image processing applications can be implemented.

Keywords: Filter bank, DWT, low power, distributive arithmetic, FIR filter, ASIC

[1] Assistant professor, Department of Electronics and Communication Engineering, Maharaja Institute of Technology, Mysore, India, Pin code: 571477.

[2] Associate Professor, Department of Electronics and Communication Engineering, Sri Jayachamarajendra Institute Of Technology, Mysore.

[3] Physcical Design engineer, Bangalore.

[4] Assistant professor, Department of Electronics and Communication Engineering,Dayanada Sagar Academy Of Technology and Management,Bangalore,pin:560082

DOI: 10.31838/ecb/2023.12.si6.324

1. INTRODUCTION

Discrete Wavelet Transform (DWT) is widely used in biometric applications for decomposing the input into sub bands localizing low frequency and high frequency components at different resolutions. DWT is computed using sub band technique that uses FIR filters with predefined filter coefficients and down sampling unit. Pair of low pass and high pass filters is used in every stage to decompose the input signal into low pass and high pass sub bands respectively. Number of stages of decomposition is based on the information required to capture information at different resolutions. For a three level symmetric decomposition of input using DWT sub bands the number of FIR filter pair required are seven. Considering the FIR filter coefficients to be of order ‘P’ and input sample length of ‘N’, computing one output for FIR filter requires P multipliers and P-1 adders. A pair of FIR filter to generate all N output from each of the FIR filter require 2NP multipliers and 2N(P-1) adders. Suitable number system is required to represent the input data and the filter coefficients in order to implement FIR filter on hardware platforms. Power dissipation and delay are the key parameters that need to be considered for efficient implementation of FIR filters for all biometric applications.

John and Chacko [1] present Differential Evolution-Ant Colony Algorithm (DE-ACO) for design of FIR filter meeting requirement of power and frequency domain specifications. In their work methods to obtain optimum solution for FIR filter coefficients is developed that requires optimum time. This algorithm is modelled and verified in ISE environment demonstrating logic correctness. Ghamkhari and Ghaznavi [2] present FIR filter architecture design based on Distributive Arithmetic (DA) method. Power and delay are optimized in their design by separating power supply of data path and controller. The design is implemented considering 65 nm CMOS technology demonstrating reduction in both dynamic and static power dissipation. Ghamkhari and Ghaznavi [3] present DA based FIR filter design architecture using gated flip-flop and shift register arrays reducing power dissipation. Low power techniques like clock gating, power gating and multi-Vdd are used to improve energy efficiency. Replacing gated flip-flop in place of conventional flip-flop has resulted in reducing power dissipation by 40%. Sundar et al. [4] have presented area efficient architecture for adaptive FIR filter based on DA method that is successfully used for filter of speech signals in hearing aids. Look-up-table and shift accumulate units are used in decimation filter which is a multiplierless architecture. Power saving of 40% is achieved compared with conventional architecture on implementation using 90nm CMOS technology. Touil et al. [5] present FIR architecture design considering Data Driven Clock Gating (DDCG) and Multi-Bit Flip-Flops (MBFF). Accumulator module is identified as the primary contributor for power and delay which is addressed by using clock gating into multi-bit techniques. The 9-tap FIR filter is implemented on Virtex-5 FPGA based on the proposed techniques and also synthesized using 0.25 micro meter CMOS technology libraries. Power saving of up to 22% is achieved with operating frequency of 100 MHz. Sumalatha et al. [6] have presented FIR filter architecture design based on Vedic multiplier-carry look ahead adder. The designed FIR filter is used as denoising unit of ECG signals and the architecture is modelled in Verilog HDL implemented on both FPGA and ASIC platform. Area, speed and power are optimized for the Urdhava Tiryakbhagyam Sutra algorithm. Odugu et al. [7] in their work present VLSI implementation of generic FIR filter using block processing and memory based multiplier for symmetric structure. DA based algorithm is used in this method and the design modelled in Verilog HDL is synthesized using 45nm CMOS technology library demonstrating reducing power dissipation to nearly 30% compared with conventional designs. Kannan and Deepa [8] [9] have designed low power FIR filter architecture which is optimized for area using modified carry save accumulator logic. The ripples that exist in FIR filter response in both pass band and stop band are eliminated. The carry input is used to choose appropriate adder logic for accumulation operation that is used in FIR filter design. Area is reduced by 24% and power dissipation by 3% compared with existing methods. In most of the designs presented DA algorithm for FIR filter implementation is used as it is multiplierless logic. The adder module that performs accumulation operation is realized using low power adder modules and low power techniques such as clock gating, power gating and multi-vdd is used. Use of FIR filter for DWT realization requires a thorough understanding of wavelet filter coefficients and accordingly suitable number system, quantization methods, arithmetic operators and data movement registers need to be designed and integrated. DA based algorithm is most suitable for

FIR filter, however it is required to customize the DA algorithm for DWT FIR filter implementation considering redundancy among filter coefficients and data movement. In this paper an improved method for FIR filter implementation considering wavelet coefficients using DA algorithm is presented. The architecture is designed and implemented on FPGA platform. Area, timing and power parameters are estimated for the proposed design and are optimized further for efficient implementation.

DWT structure

DWT algorithm comprises of multiple stages of filtering process as shown in Figure 1. Considering two level decomposition, the input signal is decomposed into four sub bands (SB1, SB2, SB3 and SB4) with two levels of sub band coding. In level 1 the pair of DWT filters L_a and H_a decompose the input signal of N samples into low pass and high pass outputs. The filtered output is down sampled by 2 to generate N/2 samples from each filter output. In level 2, the output of each of the filters in level 1 is further decomposed by pair of L_a and H_a filters to generate the level 2 sub bands. Down sampling of sub bands is carried out and the number of samples at each of the sub bands are N/4. In the reconstruction process, the decomposed sub bands are up sampled by 2 and filtered by two pair of filters represented as L_b and H_b.

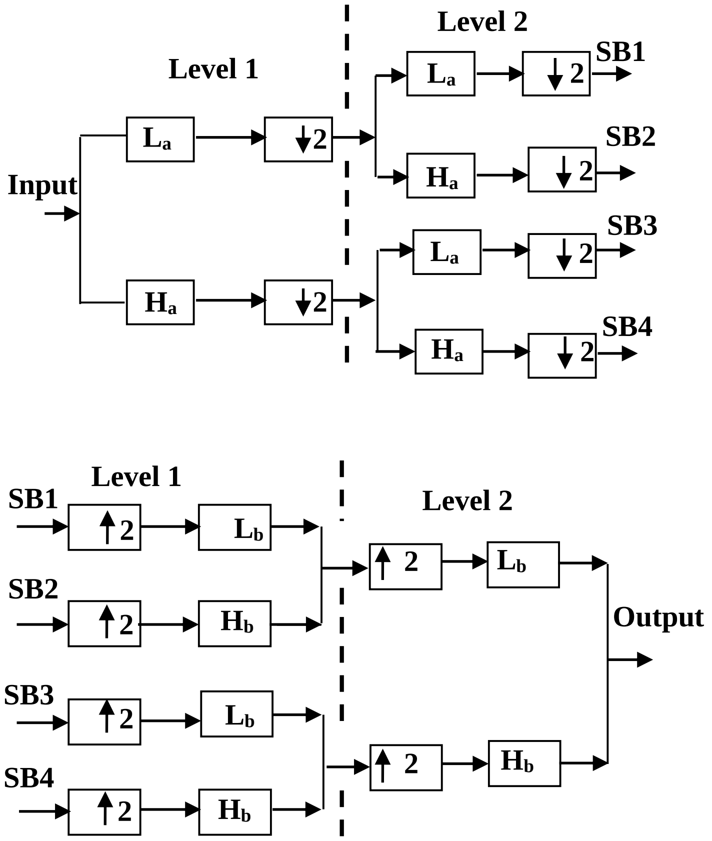

Fig. 1 DWT analysis and synthesis filter banks (with two-level decomposition)

The filtering operation is further continued in level 2 to reconstruct the original signal. For perfect reconstruction of input signal the filters need to satisfy the property as in Eq. (1) and Eq. (2).

$$L_a(-z)L_b(z) + H_a(-z)H_b(z) = 0 \quad \text{and} \quad L_a(z)H_b(z) + L_b(z)H_b(z) = 2 \qquad (1)$$

$$l_a(n) = (-1)^n h_a(n) \text{ and } h_b(n) = (-1)^{n+1} l_a(n) \qquad (2)$$

DWT structure consists of FIR filters H_0 and H_1 that are used as analysis filters and F_0 and F_1 is used as synthesis filters. Each of the filters is required to be designed optimizing area, timing and power for VLSI implementation. The filter coefficients are predefined (for example Daubechies wavelet filters, Haar filters, 9/7 filter, 5/3 filters, biorthogonal filters etc.). There are several methods for realizing FIR filters and one of the most popular methods is the DA algorithm.

DA algorithm

Considering the convolution expression for a FIR filter represented as in Eq. (3),

$$Y = \sum_{k=1}^{K} A_k X_k \tag{3}$$

Where A_k is the filter coefficients and X_k is the input, K represents the order of the filter and Y is the output of FIR filter. Expressing the input samples in 2's complement representations and considering normalized to input as ($|X_k| < 1$), where X_k is N bit number given represented as $X_k =$ $\{b_{k0}, b_{k1}, b_{k2} .....b_{k(n-1)}\}$. b_{k0} is the sign bit and X_k is expressed as in Eq. (4),

$$X_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{n-1} \tag{4}$$

Substituting Eq. (4) in Eq. (3) for X_k in 2's complement representation Eq. (5) is obtained.

$$Y = \sum_{K=1}^{K} A_k \left[-b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \right] \tag{5}$$

Simplifying Eq. (5) and rearranging the summation term Eq. (6) is obtained.

$$Y = -\sum_{K=1}^{K} \left(b_{k0} . A_k \right) + \sum_{k=1}^{k} \sum_{n=1}^{N-1} \left(A_k . b_{kn} \right) 2^{-n} \tag{6}$$

Eq. (6) represents the generic output for FIR filter considering the inputs in 2's complement representation and with filter coefficient represented as A_k. Selection of filter bank for signal and image processing application requires considering shift invariant property of the wavelet filter coefficients.

In Eq. (6), the first term is the sign bit, the second term is the partial product term. For different combinations of 'n' the weighted binary bits bkn are multiplied by the fixed coefficients Ak and the partial products are pre-computed. These pre-computed partial products are stored in the memory. The memory elements are accessed by considering the input data as the address and each of the partial product is accumulated to generate the filter output. With each stage having two filters the expression for these filters can be represented as in Eq. (7) and Eq. (8).

$$Y_{La} = -\sum_{K=1}^{10} \quad b_{k0} h_{ak} + \sum_{k=1}^{10} \quad [\sum_{n=1}^{N-1} \quad (b_{kn} L_{ak}) 2^{-n}] \tag{7}$$

$$Y_{Ha} = -\sum_{K=1}^{10} \quad b_{k0} H_{ak} + \sum_{k=1}^{10} \quad [\sum_{n=1}^{N-1} \quad (b_{kn} H_{ak}) 2^{-n}] \tag{8}$$

As there are 10 filter coefficients the expression is suitable set for computing DTCWT filter outputs. As there are 10 filter coefficients there will be 2^{10} possible partial products and as there are two filters the total number of memory units required will be 2 each of size 1024 x 10 (20480 bits). In order to reduce the number of partial products and also to increasing processing speed the modified algorithm is developed. Considering Eq. (7) and splitting into two equal terms as in Eq. (6), the number of partial products is reduced and can be reused.

$$Y_{La} = \sum_{K=0}^{4} \quad \sum_{n=1}^{N-1} \quad [(L_{ak.}b_{kn})2^{-n}]\,(term\ 1) \ + \sum_{K=5}^{9} \quad \sum_{n=1}^{N-1} \quad [(L_{ak.}b_{kn})2^{-n}]\,(term\ 2) \quad (6)$$

In the reorganized DA algorithm expression, the first term and second term have 5 filter coefficients and hence the number of partial products for each term is 2^5. The total number of storage memory bits required is 640 bits. By splitting the expression the total number of bits to store partial product is reduced by 98.43%. Table 2 presents the partial products for four filters based on the modified DA algorithm expression in Eq. (6). The block diagram for the modified DA algorithm is presented in Figure 2. The input is stored in a PISO register of depth 10. Once the data is loaded into the PISO, the LSBs from each of the 10 registers are read out to form the address for the memory unit. As the memory unit is split into two sections, the LSBs from top five registers are used as address for memory unit 1 and the LSBs from the bottom 5 registers are used as address for memory unit 2.

Table 2 LUT contents of filters

Address bits	L^1_a Filter YLa_{LUT}		H^1_a Filter YHa_{LUT}	
$A_{4/9}A_{3/8}A_{2/7}A_{1/6}A_{0/5}$	Term 1	Term 2	Term 1	Term 2
00000	0	0	0	0
00001	0	23	0	-178
00010	-23	-23	-3	178
00011	-23	0	-3	0
00100	23	3	3	-23
00101	23	26	3	-201
00110	0	-20	0	155
00111	0	3	0	-23
01000	178	3	23	-23
01001	178	26	23	-201
01010	155	-20	20	155
01011	155	3	20	-23
01100	201	6	26	-46
01101	201	29	26	-224
01110	178	-17	20	132
01111	178	6	20	-46
10000	178	0	23	0
10001	178	23	23	-178
10010	155	-23	20	178
10011	155	0	20	0
10100	201	3	26	-23
10101	201	26	26	-201
10110	178	-20	20	155
10111	178	3	20	-23
11000	356	3	46	-23
11001	356	26	46	-201
11010	333	-20	43	155
11011	333	3	43	-23
11100	379	6	49	-46
11101	379	29	49	-224
11110	356	-17	43	132
11111	356	6	43	-46

The modified architecture in Figure 2 requires two memory elements each of depth 32, two accumulators of 11-bit and two adders of 12-bit. Each partial product read out from the Look Up Tables (LUT) are accumulated in the accumulator section and the final output is generated at the output of the summer. As the input data width is 9, the partial products are read out 9 times for different combinations of address bits and the accumulator performs operation for 9 times and in the 10th clock the final output is generated at the output of the summer. The latency of the modified DA structure is 20 clocks (10 clocks for data loading into PISO, 10 clocks for data read and accumulation).

It is observed that the contents of YLa_{LUT1}, YHa_{LUT1}, YLb_{LUT1}, YHb_{LUT1} from memory locations 10000 to 11111 is observed to be 178, 23, 23, -178 higher than memory contents from locations 00000 to 01111. It is also observed that the memory contents of YLa_{LUT2}, YHa_{LUT2}, YLb_{LUT2}, YHb_{LUT2} from memory locations 00000 to 01111 is similar to the contents in memory locations 10000 to 11111. Based on these observations the reduced memory DA algorithm is designed.

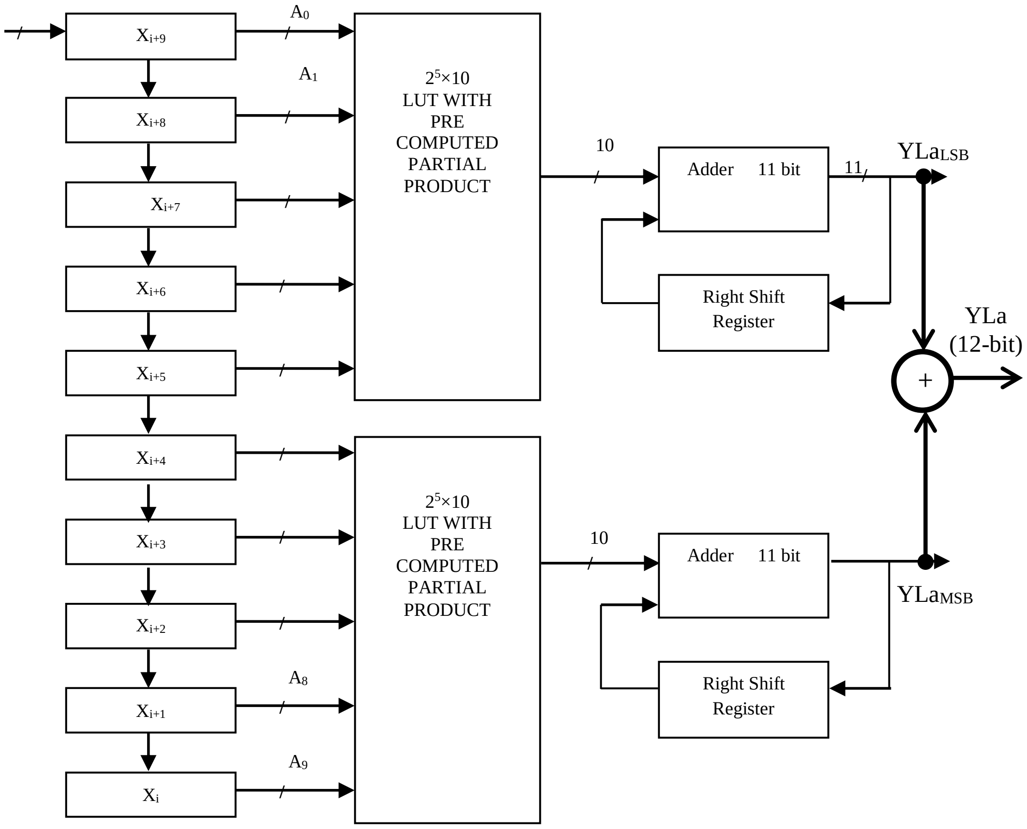

Fig. 2 Modified DA structure for YLa filter

Memory efficient DA Architecture

Table 3 presents the LUT content for four filters considering only term 1 of expression presented in Eq. (6). The LUT contents provided are regrouped into two components considering the MSB address A4. If address A4 is 0 the contents of LUT areYLaLUT1, YHa_{LUT1}, YLb_{LUT1} and YHa_{bUT1} if the address A4 is 1 the contents of LUT are 178 + YLa_{LUT1}, 23 + YHa_{LUT1}, 23 + YLb_{LUT1} and 178 + YHa_{bUT1}. The address bits A_3 A_2 A_1 A_0 are used to access the LUT contents and the LUT depth is 16.

Table 3 Modified LUT contents for memory efficient DA (Term 1)

	L^1_a Filter Term 1		H^1_a Filter Term 1		L^1_b Filter Term 1		H^1_b Filter Term 1	
Address bits	A_4=0	A_4=1	A_4=0	A_4=1	A_4=0	A_4=1	A_4=0	A_4=1
A_3 A_2 A_1 A_0	YLa_{LUT1}	178 + YLa_{LUT1}	YHa_{LUT1}	23 + YHa_{LUT1}	YLb_{LUT1}	23 + YLb_{LUT1}	YHb_{LUT1}	178 + YHb_{LUT1}
0000	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0

0010	-23	-23	-3	-3	3	3	-23	-23
0011	-23	-23	-3	-3	3	3	-23	-23
0100	23	23	3	3	3	3	-23	-23
0101	23	23	3	3	3	3	-23	-23
0110	0	0	0	0	6	6	46	46
0111	0	0	0	0	6	6	-46	-46
1000	178	178	23	23	23	23	178	178
1001	178	178	23	23	23	23	178	178
1010	155	155	20	20	26	26	155	155
1011	155	155	20	20	26	26	155	155
1100	201	201	26	26	26	26	155	155
1101	201	201	26	26	26	26	155	155
1110	178	178	20	20	29	29	132	132
1111	178	178	20	20	29	29	132	132

The DA structure for Term 1 of YLa filter is presented in Figure 3. The five input registers store the input data and the LSBs of these registers are connected to the LUT address bits. The LSB of top register is considered as address bit A4 which is connected to the multiplexer select line. At every clock the LSBs of all four registers are used to read out the LUT content. The output of LUT is accumulated according to the combinations of address bits and the accumulated content is sent to the output summer. The summer circuit performs addition of the accumulated output along with a constant number that is read from the output of 2:1 multiplexer. For YLa filter the constants are 0 and 178 which is used in the summer circuit to generate the final output and is stored in output register Reg1.

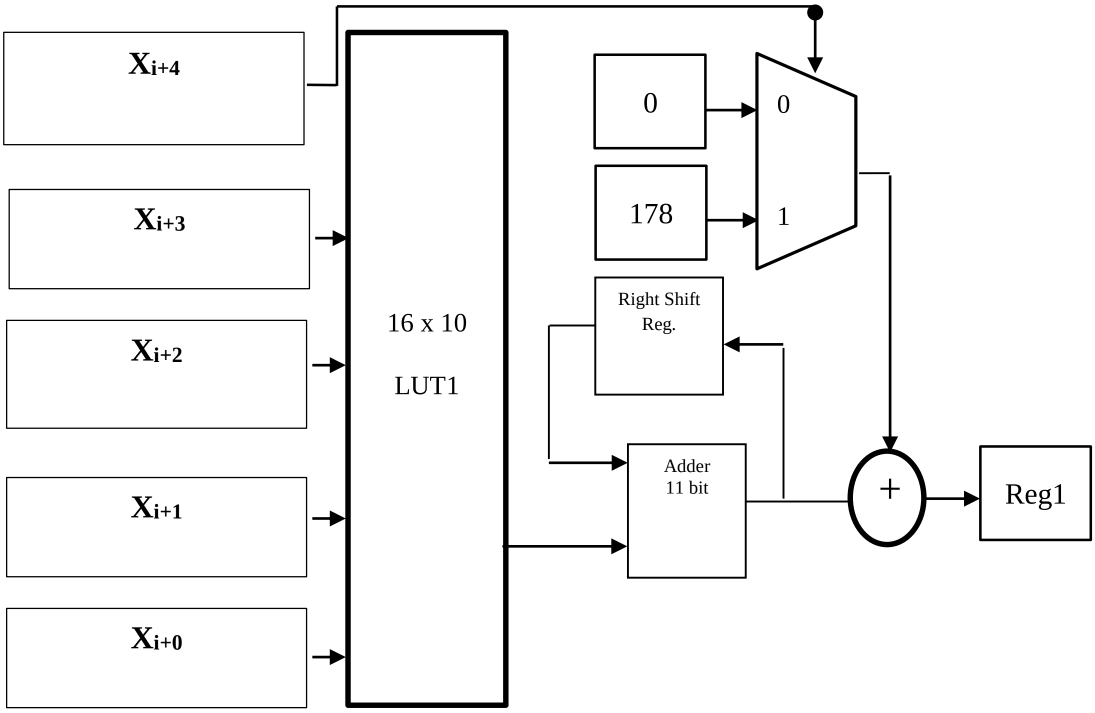

Fig. 3 Memory efficient DE structure for YLa term 1 filter

DA structure for YHa term 1 filter is presented in Figure 4 and the constants are 0 and 23. Similarly the DA structure for YLb and YHb filters are designed.

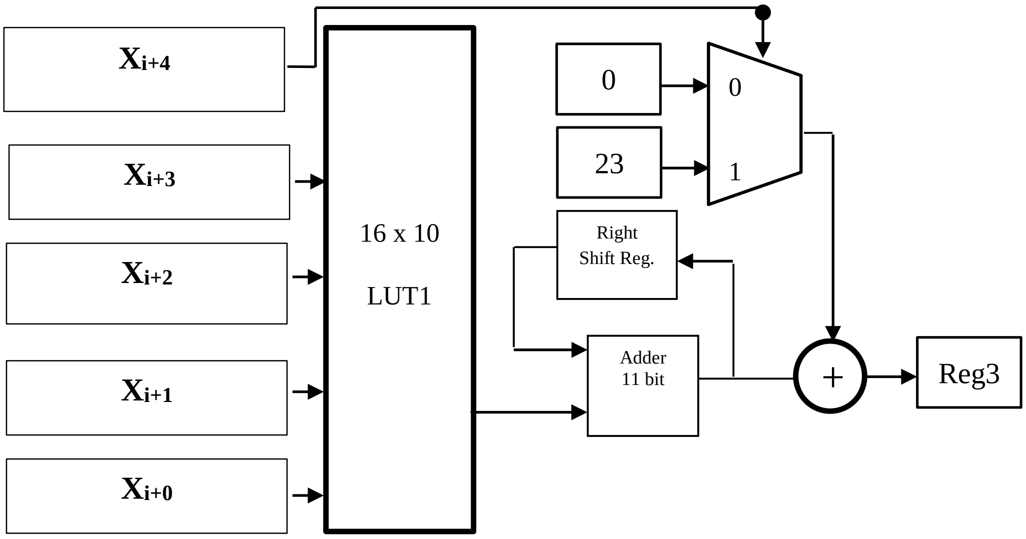

Fig. 4 Memory efficient DA structure for YHa term 1 filter

The DA structure for term 2 for all four filters is deigned considering the LUT contents presented in Table 4. For the address bits A_3 A_2 A_1 A_0 the LUT contents remain same for both possible conditions of MSB address bit A_4. Considering the LUT contents for all four filters as indicated in Table 5 the LUT size will be 16. Upon observation of LUT data the partial products of LUT for address bits 0000 to 0111 and 1000 to 1111 are repeated.

Table 4 Modified LUT contents for memory efficient DA (Term 2)

	L^1_a Filter Term 2		H^1_a Filter Term 2		L^1_b Filter Term 2		H^1_b Filter Term 2	
Address bits	A_4=0	A_4=1	A_4=0	A_4=1	A_4=0	A_4=1	A_4=0	A_4=1
A_3 A_2 A_1 A_0	YLa_{LUT2}	YLa_{LUT2}	YHa_{LUT1}	YHa_{LUT1}	YLb_{LUT1}	YLb_{LUT1}	YHb_{LUT1}	YHb_{LUT1}
0000	0	0	0	0	0	0	0	0
0001	23	23	-178	-178	0	0	23	23
0010	-23	-23	178	178	178	178	23	23
0011	0	0	0	0	178	178	46	46
0100	3	3	-23	-23	23	23	3	3
0101	26	26	-201	-201	210	210	26	26
0110	-20	-20	155	155	210	210	26	26
0111	3	3	-23	-23	379	379	49	49
1000	3	3	-23	-23	-23	-23	-3	-3
1001	26	26	-201	-201	-23	-23	20	20
1010	-20	-20	155	155	155	155	20	20
1011	3	3	-23	-23	155	155	43	43
1100	6	6	-46	-46	0	0	0	0
1101	29	29	-224	-224	178	178	23	23
1110	-17	-17	132	132	178	178	23	23
1111	6	6	-46	-46	356	356	46	46

Considering the redundancy in the LUT content further modification is carried out and the contents of LUT are reduced as presented in Table 5. The address bits A_2 A_1 A_0 are used to access the LUT contents and the address bits A_4 A_3 are used to enable the constant number to be added with the accumulated output at the summer circuit. The LUT contents for YLa and YHa filter term 2 is shown in Table 5, similarly the LUT content for YLb and YHb can be identified.

Table 5 Reduced memory DA contents for term 2 filters

	L^1_a Filter Term 2		L^1_a Filter Term 2	
Address bits	A_4 A_3=00	A_4 A_3=01	A_4 A_3=10	A_4 A_3=11
A_2 A_1 A_0	YLa_{LUT2}	3 + YLa_{LUT2}	YHa_{LUT1}	3 + YHa_{LUT1}
000	0	0	0	0
001	23	23	23	23
010	-23	-23	-23	-23
011	0	0	0	0
100	3	3	3	3
101	26	26	26	26
110	-20	-20	-20	-20
111	3	3	3	3

Figure 5 presents the reduced memory DA structure. The top two registers that from the address bits A4 and A3 are not connected to the LUT address bits. The bottom three registers LSBs form the address to the LUT. The depth of LUT is 8 and the LUT content is read our every clock is accumulated at the accumulator module. The accumulated data is given to the summer to add the corresponding constant depending upon the status of address bit A3. The summer output is stored in the register Reg3. The final output of the filter is generated by summing the output of Reg1 and Reg3. Direct implementation of filter using DA algorithm will require 10 input registers, LUT of size 1024 x 10 and 11-bit accumulator. The total propagation delay will be $T_{LUT} + T_{ACC}$ that represent the delay of LUT data read out and delay in accumulator. With four filters in the first stage the total number of sub modules required are presented in Table 6 and compared with the optimized structure proposed.

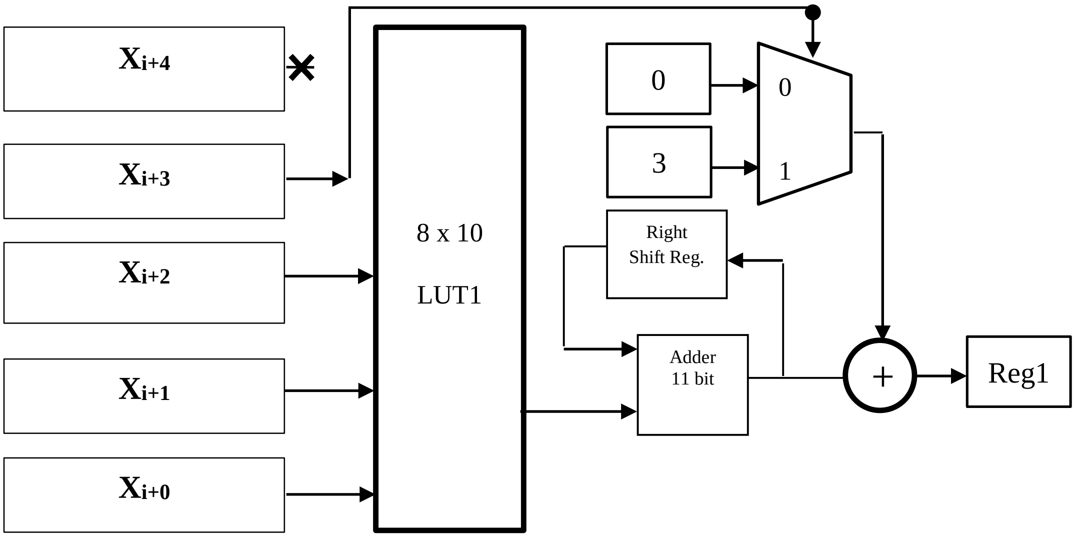

Fig. 5 Reduced memory DA structure for term 3 filter

Table 6 Comparison of DA methods

Implementation method	Filter	LUT Size	No. of adders	No. of accumulators	Critical path (T)	Clock delay	
						Latency	Throughput
Direct DA	YLa	1024	0	1	T_{LUT} $+T_{ACC}$	21	12
	YHa	1024	0	1		21	12
	YLb	1024	0	1		21	12

	YHb	1024	0	1		21	12
Split DA	YLa	32	1	2	$T_{LUT} + T_{ACC} + T_{ADD}$	22	13
	YHa	32	1	2		22	13
	YLb	32	1	2		22	13
	YHb	32	1	2		22	13
Proposed DA	YLa	24	3	2	$T_{LUT} + T_{ACC} + 2T_{ADD}$	23	14
	YHa	24	3	2		23	14
	YLb	24	3	2		23	14
	YHb	24	3	2		23	14

The advantages of proposed method are in terms of memory size of LUTs. The proposed method requires two LUTs per filter of size 16 x 10 and 8 x 10 (10 is bit width of LUT contents). Comparing with direct method of implementation the savings in memory size is 97.65% per filter. The number of adders and accumulators are increased by 3 and 1 compared with direct implementation. The critical path is increased by $2T_{ADD}$, latency and throughput is increased by 2 clock cycles.

Adder design

In this work, three different adders are used and the propagation delay of all the three adders is estimated and is presented in Table 7. The accumulator unit is also designed and estimated for its power dissipation. Considering the sub blocks, the HDL model for the proposed architecture is modeled and is simulated for its logic correctness.

Table 7 Power consumption for different adders per unit (nW)

ADDER	POWER (nW)
RIPPLE CARRY ADDER	602.95
CLA	567.45
CSA	551.55

ASIC implementation

Logic synthesis is the process of converting a design description written in a hardware description language such as Verilog or VHDL into an optimized gate-level netlist mapped to a specific technology library. The YOSYS is used for Logical Synthesis, both the Design Rule Constraints and Optimization constraints are applied. The target and the like library chosen, the technology libraries contain the information about the characteristics and functions of each cell. Logic library contain the information about the library group, library level attributes, environment description. The Design Rule Constraints are applied to the design using the Technology Library. The Optimization Constraints are defined by the designer. Timing constraints, Maximum area, and Minimum porosity are the different optimization constraints. Timing Constraints are given the highest priority and is the one used for optimization in this design. In addition to the setup and hold analysis, the STA signoff tool in Qflow performs recovery, removal, gate setup, gate hold and min pulse width checks. The Endpoint slack histogram from primetime provides a graphical view of the distribution of various paths w.r.t. the slack. The two end points of slack (min and max) are considered and divided into some bars for grouping.

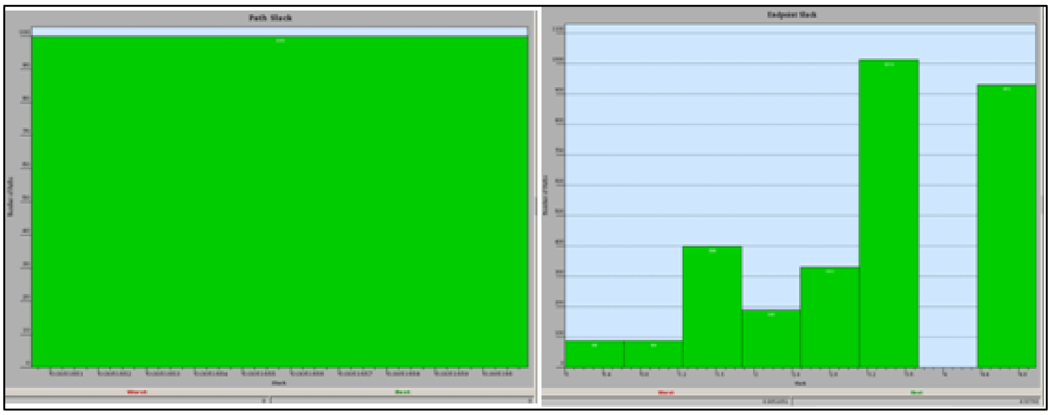

Figure 6 Path slack and endpoint slack Histogram of proposed filter bank

The bars are representing path of the design (Figure 6). Green bar represents the path which met the timing and the red bar represents the path which did not meet the timing. The design of filter bank has seven paths. All the paths have green bars which represents all the paths of the filter bank design met the timing. The sixth bar from the left is representing that, path has maximum delay. The first bar and second bar from the left is representing that, path has minimum delay (Figure 7).

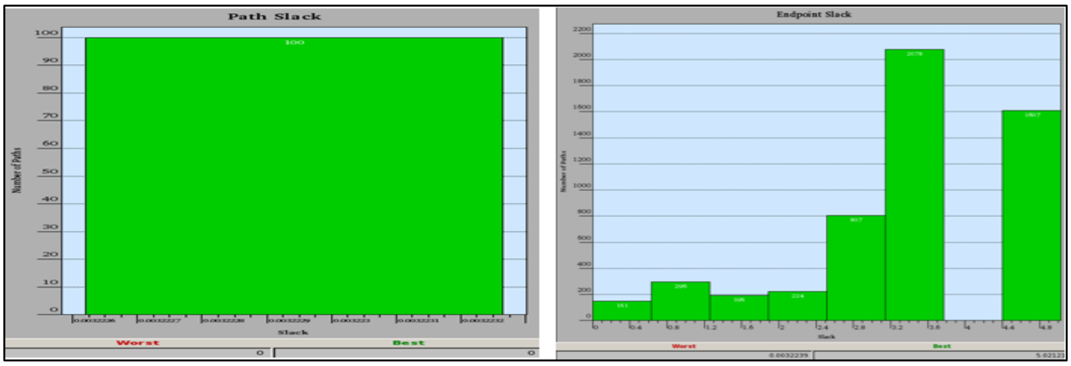

Figure 7 Path slack and endpoint slack Histogram of IDTCWT

The bars are representing path of the design. Green bar represents the path which met the timing and the red bar represents the path which did not meet the timing. The design of IDTCWT has seven paths. All the paths have green bars which represents all the paths of the IDTCWT design met the timing. The sixth bar from the left is representing that, path has maximum delay. The first bar from the left is representing that, path has minimum delay.

Table 8 Synthesis results of filter bank structure

Report	**YOSYS**	**YOSYS (post DFT)**	**PT (post DFT)**	**PD**
Area(sq.µmm)	49756.95	60634.73	-	75782.81
Power(nW)	5.2687	10.2465	6.673	10.7689
Setup Timing	4.53	4.57	6.07	0.5
Setup slack	0.00	0.01	0.02	0.197
Hold Timing	0.45	0.37	0.30	0.25
Hold Slack	0.00	0.00	0.32	0.214

From the PD flow it is estimated that the power dissipation is limited to less than 20 mW for both DTCWT and IDTCWT which is found to be higher than the estimation carried out during synthesis process.

Table 9 Physical design power results of DTCWT

Power	**PD**
I/O Net switching Power(nW)	0.05285
Total switching Power(nW)	2.34769
Total Short circuit Power(nW)	5.55174
Total internal Power(nW)	2.65167
Total leakage Power(nW)	0.2177
Total Power(nW)	10.7689

Table 9 present the power dissipation of filter bank structure computed after physical design process in Qflow environment. Table 10 summarizes and compares the ASIC implementation results of reduced order and zero padded based systolic array architecture design for DTCTW.

Table 11 Comparison of DTCWT architecture

ASIC implementation parameters	**Direct Implementation [11]**	**Improved method [12]**	**Proposed Design**
No. of wires	358	514	452
No. of memory	0	2	10
Standard cell Area	860000000	7400000000	670000000
Power	27.266193 nW	18.92333 nW	10.7689 nW
Maximum delay path	3543.78 ps	3332.2 ps	2678.91 ps
Minimum delay path	472.527 ps	388.22 ps	354.38 ps
Cell Height	960000	960000	960000
Cell Width	430000	360000	240000
Aspect Ratio	0.75	0.75	0.75

2. CONCLUSION

In this paper, ASIC implementation of filter bank structure that is required for computing DWT is presented. The designed architecture implemented on ASIC considering 0.45 micrometre technology. The simulation results are verified and the functional verification is carried out. ASIC synthesis is carried out using YOSYS and STA is also performed. Floor planning, placement and routing is carried out in Qflow environment. From the reports generated power is estimated before physical design and after physical design. From the results obtained, proposed design is best suitable for low power and high speed applications.

Acknowledgement: The authors would like to thank Dr. Cyril Prasanna Raj P. for providing support in carrying out experimental work.

3. REFERENCES

1. Tintu Mary John and Shanty Chacko, Efficient VLSI architecture for FIR filter design using modified differential evolution ant colony optimization algorithm, Circuit World, ISSN: 0305-6120 ()
2. Seyede Fatemeh Ghamkhari, Mohammad Bagher Ghaznavi-Ghoushchi, A power–performance partitioning approach for low-power DA-based FIR filter design with emphasis on datapath and controller, International Journal of Circuit Theory and Applications, Volume 50, Issue 2 p. 427-447

3. Seyede Fatemeh Ghamkhari, Mohammad Bagher Ghaznavi-Ghoushchi, A New Low Power Schema for Stream Processors Front-End with Power-Aware DA-Based FIR Filters by Investigation of Image Transitions Sparsity, Circuits, Systems, and Signal Processing (2021) 40:3456–3478
4. Praveen Sundar, P.V., Ranjith, D., Karthikeyan, T. *et al.* Low power area efficient adaptive FIR filter for hearing aids using distributed arithmetic architecture. *Int J Speech Technol* 23, 287–296 (2020)
5. Lamjed Touil, Abdelaziz Hamdi, Ismail Gassoumi, Abdellatif Mtibaa, "Design of Low-Power Structural FIR Filter Using Data-Driven Clock Gating and Multibit Flip-Flops", Journal of Electrical and Computer Engineering, vol. 2020, Article ID 8108591, 9 pages, 2020
6. M. Sumalatha, P.V. Naganjaneyulu, K. Satya Prasad, Low power and low area VLSI implementation of vedic design FIR filter for ECG signal de-noising, Microprocessors and Microsystems, Volume 71, 2019, 102883, ISSN 0141-9331
7. Venkata Krishna Odugu; C. Venkata Narasimhulu; K. Satya Prasad, Implementation of Low Power Generic 2D FIR Filter Bank Architecture Using Memory-based Multipliers, Journal of Mobile Multimedia, Vol. 18, Issue 3, 2022
8. L. Mohana Kannan and D. Deepa, A design of low power and area efficient FIR filter using modified carry save accumulator method, Turkish Jouranl of Computer and Mathematics Education, Vol. 12, No. 7, pp. 1735-1750, 2021
9. B. N. Mohan Kumar and H. G. Rangaraju, Design and implementation of pervasive DA based FIR filter and feeder register based multiplier for software defined radio networks, International Journal of Pervasive Computing, ISSN: 1742-7371
10. John D. Villasenor, Benjamin Belzer, and Judy Liao, Wavelet filters evaluation for image compression, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 4, NO. 8, AUGUST 1995
11. Dilip R, Bhagirathi V. (2013) Image Processing Techniques for Coin Classification Using LabVIEW. OJAI 2013, 1(1): 13-17 Open Journal of Artificial Intelligence DOI:10.12966/ojai.08.03.2013
12. Naveen Mukati, Neha Namdev, R. Dilip, N. Hemalatha, Viney Dhiman, Bharti Sahu, Healthcare Assistance to COVID-19 Patient using Internet of Things (IoT) Enabled Technologies,Materials Today: Proceedings,2021,https://doi.org/10.1016/j.matpr.2021.07.379. ISSN214-7853,
13. Mr. DILIP R, Dr. Ramesh K. B. (2020). Development of Graphical System for Patient Monitoring using Cloud Computing. International Journal of Advanced Science and Technology, 29(12s), 2353 - 2368.
14. Mr. Dilip R, Dr. Ramesh K B ."Design and Development of Silent Speech Recognition System for Monitoring of Devices", Volume 7, Issue VI, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: , ISSN : 2321-9653
15. R. Dilip, Y. D. Borole, S. Sumalatha and H. Nethravathi, "Speech Based Biomedical Devices Monitoring Using LabVIEW," 2021 9th International Conference on Cyber and IT Service Management (CITSM), 2021, pp. 1-7, doi: 10.1109/CITSM52892.2021.9588853.
16. Pandey, J.K. et al. (2023). Investigating Role of IoT in the Development of Smart Application for Security Enhancement. In: Sindhwani, N., Anand, R., Niranjana Murthy, M., ChanderVerma, D., Valentina, E.B. (eds) IoT Based Smart Applications. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-031-04524-0_13.
17. Dilip, R., Samanvita, N., Pramodhini, R., Vidhya, S.G., Telkar, B.S. (2022). Performance Analysis of Machine Learning Algorithms in Intrusion Detection and Classification. In: Balas, V.E., Sinha, G.R., Agarwal, B., Sharma, T.K., Dadheech, P., Mahrishi, M. (eds) Emerging Technologies in Computer Engineering: Cognitive Computing and Intelligent IoT. ICETCE 2022. Communications in Computer and Information Science, vol 1591. Springer, Cham. https://doi.org/10.1007/978-3-031-07012-9_25.
18. Pratik Gite, Anurag Shrivastava, K. Murali Krishna, G.H. Kusumadevi, R. Dilip, Ravindra Manohar Potdar, under water motion tracking and monitoring using wireless sensor network

and Machine learning,Materials Today: Proceedings,2021,ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2021.07.283.

19. Anurag Shrivastava, Chinmaya Kumar Nayak, R. Dilip, Soumya Ranjan Samal, Sandeep Rout, Shaikh Mohd Ashfaque, automatic robotic system design and development for vertical hydroponic farming using IoT and big data analysis,Materials Today: Proceedings,2021,ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.07.294.

20. Gupta, N. ., Janani, S. ., R, D. ., Hosur, R. ., Chaturvedi, A. ., & Gupta, A. . (2022). Wearable Sensors for Evaluation Over Smart Home Using Sequential Minimization Optimization-based Random Forest. International Journal of Communication Networks and Information Security (IJCNIS), 14(2), 179–188. https://doi.org/10.17762/ijcnis.v14i2.5499