Original scientific paper

https://doi.org/10.33180/InfMIDEM2021.204

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 51, No. 2(2021), 119 – 134

Low Power Area Optimum Configurable 160 to 2560 Subcarrier Orthogonal Frequency Division Multiplexing Modulator-Demodulator Architecture based on Systolic Array and Distributive Arithmetic Look-Up Table

Girish Nanjareddy[1], *Veena Mysuru Boregowda*[2], *Cyril Prasanna Raj*[3]

[1]*Visvesvaraya Technological University, BMS College of Engineering, Department of Electronics, Bengaluru, Karnataka, India.*
[2]*BMS College of Engineering, Department of Electronics, Bengaluru, Karnataka, India.*
[3]*CIT, Department of Electronics, Bengaluru, Karnataka, India.*

Abstract: In this work, configurable and reusable Dual-Tree Complex Wavelet Transform (DTCWT) Orthogonal Frequency Division Multiplexing (OFDM) modulator-demodulator based on Optimum Systolic Array (OSA) and Modified Distributive Arithmetic (MDA) algorithm is designed for low power underwater MODEM applications. The DTCWT-Inverse Dual-Tree Complex Wavelet Transform (IDTCWT) filters are designed considering integer 10-tap Q shift filter coefficients that are quantized and rounded off to achieve symmetry among filter coefficients. Multi stage DTCWT structure is used to perform 2560 subcarrier modulation and demodulation using MDA and OSA modules. The OSA structure is designed with optimum placement of Processing Elements (PE) and the MDA structure is designed to compute two filter outputs per module with Look-Up Table (LUT) of depth 8. The 2560 modulation is carried out using folded pipelined structure that comprises of two fold and four fold configurable modules. The reusable pipelined folded OFDM modulator-demodulator is implemented on Virtex-5 FPGA and operates at a maximum frequency of 248 MHz occupying less than 15% of FPGA resources and consumes less than 1.33 W power.

Keywords: DTCWT; optimum systolic array; folded pipelined structure; FPGA design; underwater MODEM; low power architecture

Nastavljiv modulator-demodulator s frekvenčnim multipleksiranjem s 160 do 2560 ortogonalnimi podnosilci, zasnovan na arhitekturi sistoličnih polj s porazdeljeno vpogledno tabelo

Izvleček: V članku je predstavljen nastavljiv modulator-demodulator s frekvenčnim multipleksiranjem z ortogonalnimi podnosilci, zasnovan na arhitekturi sistoličnih polj s porazdeljeno vpogledno tabelo (DTCWT). Namenjen je uporabi v podvodnih sistemih nizkih moči MODEM. Zasnovani so z upoštevanjem kvantiziranih in zaokroženih Q-shift koeficientov filtra za doseganje simetrije med koeficienti. Večstopenjska DTCWT struktura modulira in demodulira 2560 podnosilcev z uporabo MDA is OSA modulov. OSA struktura je zasnovana z optimalno pozicijo procesnih elementov (PE). MDA struktura računa dva izhoda filtrov z uporabo vpogledne tabele v osmih nivojih. Modulacija uporablja dva- ali štiri-kratno zloženo strukturo v nastavljivih modulih. OFDM modulator-demodulator je uporabljen v Virtex-5 FPGA okolju in deluje pri največji frekvenci 248 MHz, pri čemer zasede manj kot 15% procesorske moči FPGA in porabi manj kot 1.33 W moči.

Ključne besede: DTCWT; optimalni sistolični niz; zložena cevovodna konstrukcija; zasnova FPGA; podvodni MODEM; arhitektura z majhno porabo

** Corresponding Author's e-mail: giribms17@gmail.com*

1 Introduction

Inverse DTCWT performs OFDM modulation at the transmitter and DTCWT is used to retrieve symbols at the receiver. DTCWT decomposition is like Discrete Wavelet Transform (DWT) but with decomposition structures of real and imaginary trees. Hardware implementation of DTCWT is twice complex than DWT implementation. Goalie et al. [1] use DWT based subcarrier modulation for OFDM. Shift variance in DWT energy levels introduce additional errors in OFDM subcarrier modulation and demodulation. Replacing IDWT-DWT with IDTCWT-DTCWT improves BER performances in OFDM. Po-Cheng Wu et al. [2] proposes DWT architectures using arithmetic blocks based on linear convolution property of the wavelet filters. Architectures for DWT implementation on FPGA platform optimizing area, speed and power have been reported, that can be used for DTCWT implementation. Direct mapped architecture, folded architecture, multiply and accumulate based programmable architecture, flipping architecture, recursive architecture, lifting based architectures, dual scan architecture and DSP type architecture are used for DWT implementation reported in [3]-[6]. Grzeszczak et al. [7] have proposed single stage systolic array architecture proving improvement in throughput and latency with the large area occupied by multipliers on FPGA platform. Chao Cheng et al. [8] have proposed a high-speed single stage architecture based on hardware efficient parallel finite impulse response (FIR) filter structures for the DWT calculation. These structures differ in terms of size of arithmetic unit, on-chip memory, cycle period and average calculation time (ACT) [9]. Chengjun Zhang et al. [10] proposed a scheme for the design of pipeline architecture for a fast calculation of the DWT on Xilinx FPGA running at a maximum frequency of 200MHz and power dissipation of 1005mW. Xin Tian et al. [11] have proposed a line based scanning scheme and a folded architecture for the calculation of multilevel 2-D DWT level-by-level. Yeong-Kang Lai et al. [12] have used parallel data access scheme to avoid line buffers to implement the reversal mechanism in the folded design using both parallel and pipeline processing logic. Chih-Chi Cheng et al. [13] have proposed a convolution based recursive architec-ture for 2-D DWT using 9/7 filters, where the throughput rate is increased in a controlled manner that requires large storage space and decomposition time. Design of DTCWT architecture is like DWT architectures reported in literature Divakar et al. [14] that requires a large storage area.

DTCWT filter structure as presented by Kingsbury requires 10-tap filter structure with four filter banks every stage. Systolic array algorithm for data processing is reported to achieve high throughput, reduced potential with reusability logic presented in Divakar et al. [15]-[16]. The DTCWT architecture reported by Poornima et al. [17] presents the design of systolic array architecture using multiplexed distributive arithmetic algorithm for image decomposition. The architecture is implemented on FPGA and operates at a maximum frequency of 300 MHz consuming less than 10 mW of power and 12% of FPGA resources. Most of the DTCWT architectures reported in the literature is for image processing and the DTCWT architecture for performing OFDM modulation presented in this work is the first of its kind implemented on FPGA for 512 symbols subcarrier OFDM. In this paper, pipelined and optimized filter structures are designed for computing both DTCWT and inverse DTCWT based on OSA logic for OFDM.

2 DTCWT Architecture design

In this work a generalized 2048 stage architecture is designed that is customized for both implementations of DTCWT as well as IDTCWT. Figure 1 presents the 7-level IDTCWT structure. The input data x_i^* (i=0,1,2,3..7….15) representing complex symbols generated from the QAM or QPSK modulators are processed by the 7-level IDTCWT structure performing OFDM represented as X'R and X'I. The DTCWT structure at the receiver performs demodulation to generate the symbols x_i^* either from received data X'R and X'I. The DTCWT filter coefficients are 10-tap Q shift filters represented as {H'00a, H'01a, H'10b, H'11b} for the first stage and all subsequent stages, and {H'0a, H'1a, H'0b, H'1b} for last stage of filtering. In every stage, there are four filters (or two pairs of filters representing real and two for imaginary part). The inputs $\left\{x_0^*, x_1^*, x_8^*, x_9^*\right\}$ are processed by the first stage filters {H'00a, H'01a, H'10b, H'11b}, with 10 filter coefficients the number of arithmetic operations per output will be 10 multiplications and 9 additions. Considering 7 stages and four filters in the stage the number of arithmetic operations will be 280 multiplications and 252 additions per output. Considering 2048 levels the total number of multiplications and additions are 81920 and 73728, respectively. The 10-tap filters require 16-bits for representation and the arithmetic operations need to carry out using floating point logic. To reduce the computation complexity fixed point integer logic is used and the 10-tap filters (N-tap) are scaled and rounded to an integer by multiplying with 64. The integer filter coefficients for the first stage and the last stage of DTCWT and IDTCWT are shown in Table 1. The filters Is1 and Is2 represent the integer filter coefficients for the first stage and all succeeding stage of DTCWT filters. Ins1 is the filter coefficient for the last stage and Ins2 is the filter coefficient

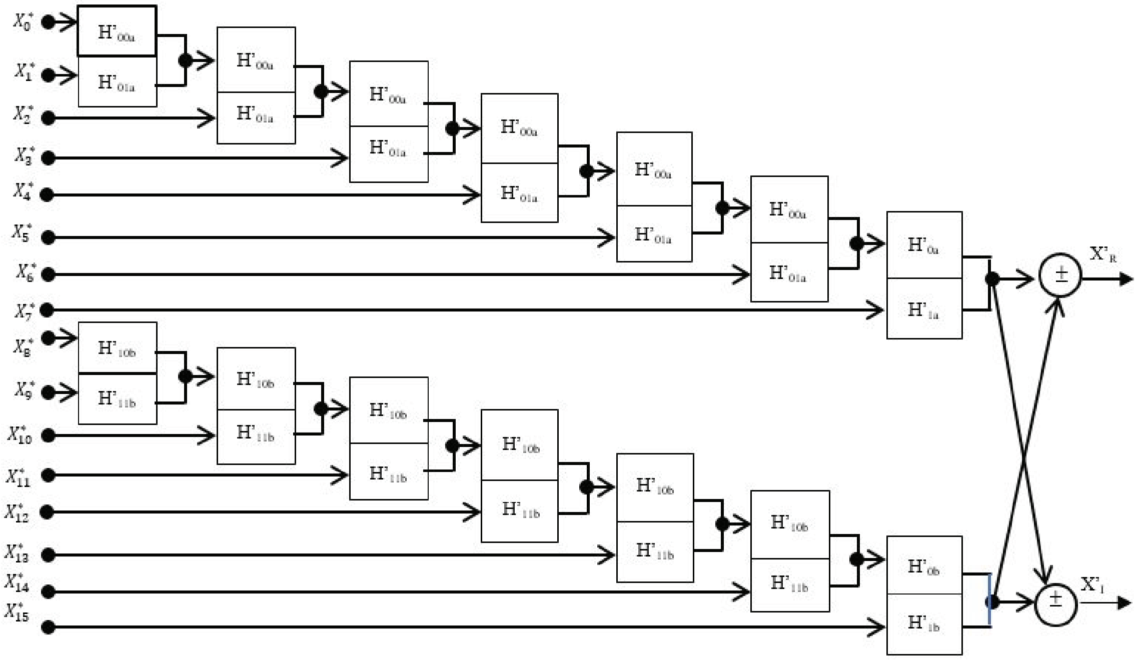


Figure 1: Seven stage IDCWT OFDM modulation for 16 symbols.

for the 1st stage of IDTCWT. The DTCWT coefficients Is2 is equal to IDTCWT coefficients Ins2 represented by {ILa, IHb, ILb, IHb}. The filter coefficients Is1 and Ins1 are related as h (n) = -h (N-1), n = 0, 1, 2, 3….9. The filter outputs generated by {ILa, IHb, ILb, IHb} are represented by { y_0^0, y_0^1, y_0^2, y_0^3 } and mathematically expressed as in Eq. (1), where a0 to a9 are the filter coefficients.

$$y_0^i = \{a_0x_0 + a_1x_1 - a_1x_2 + a_2x_3 + a_2x_4 + a_1x_5 + \\ + a_1x_6 - a_3x_7 + a_3x_8 + a_0x_9\} \tag{1}$$

The filter coefficients of Ins2 {±7, ± 38, ±49} are approximated to {±6, ±44, ±44} to bring symmetry in the filter coefficients with an error of {±1, ±6, ±5} respectively. Approximated filter coefficients are indicated in brackets in Table 1. Grouping the terms considering common terms and rearranging Eq. (1) is reduced to Eq. (2) which represents the first filter output y_0^0 with filter coefficients ILa.

$$y_0^0 = \{a_0(x_0+x_9)+a_1(x_1+x_2)+a_2(x_3+x_4)+ \\ +a_1(x_5+x_6)+a_3(x_7+x_8)\} \tag{2}$$

In Eq. (2), the arithmetic operations required are five multiplications and five additions for input terms and 4 adders for partial product terms. With the rearrangement of common terms, the number of multipliers is reduced to 5 from 10 and the number of adders re-

Table 1: Low pass and high pass filter coefficients.

n	DTCWT								IDTCWT							
	Is1				Is2				Ins1				Ins2			
	IL$_a$	IH$_a$	IL$_b$	IH$_b$	IL$_a$	IH$_a$	IL$_b$	IH$_b$	IIL$_a$	IIH$_a$	IIL$_b$	IIL$_a$	IIL$_a$	IIH$_a$	IIL$_b$	IIL$_a$
0	0	0	1	0	2	0	0	-2	0	0	0	-1	2	0	0	-2
1	-6	-1	1	0	0	0	0	0	1	-6	0	1	0	0	0	0
2	6	1	-6	-6	-6	-6	-6	6	1	-6	-6	6	-6	-7(-6)	-7(-6)	6
3	45	6	6	-6	15	0	0	15	-6	45	6	6	15	0	0	15
4	45	6	45	45	44	44	44	-44	6	-45	45	-45	49(44)	38(44)	38(44)	-49(-44)
5	6	-45	45	-45	44	-44	44	44	45	6	45	45	38(44)	-49(-44)	49(44)	38(44)
6	-6	45	6	6	0	15	15	0	45	6	6	-6	0	15	15	0
7	1	-6	-6	6	-6	6	-6	-6	6	1	-6	-6	-7(-6)	6	-6	-7(-6)
8	1	-6	0	1	0	0	0	0	-6	-1	1	0	0	0	0	0
9	0	0	0	-1	0	-2	2	0	0	0	1	0	0	-2	2	0

quired is 9. Considering Eq. (2), the total delay in computing every output sample will be 3 clock cycles (1 clock for adding input data x, 1 clock for multiplication of filter coefficient with input data and 1 clock for adding all the multiplied terms). Similarly, the filter outputs $y_0^1,\ y_0^2,\ y_0^3$ for the remaining four filters are expressed as in Eq. (3).

$$\left.\begin{aligned}
y_0^1 &= a_0\left(x_0+x_9\right)+a_3\left(x_1+x_2\right)+a_4\left(x_3+x_4\right)+\\
&\quad a_2\left(x_5+x_6\right)+a_1\left(x_7+x_8\right)\quad 3(a)\\
y_0^2 &= a_3\left(x_0+x_9\right)+a_1\left(x_1+x_2\right)+a_2\left(x_3+x_4\right)+\\
&\quad a_1\left(x_5+x_6\right)+a_4\left(x_7+x_8\right)\quad 3(b)\\
y_0^3 &= a_0\left(x_0+x_9\right)+a_1\left(x_1+x_2\right)+a_2\left(x_3+x_4\right)+\\
&\quad a_1\left(x_5+x_6\right)+a_3\left(x_7+x_8\right)\quad 3(c)
\end{aligned}\right\}\qquad(3)$$

From Eq. (2), Eq. (3a) and comparing the terms in Eq. (3b) and 3(c) the accumulated terms of input data are common. Eq. (2) and Eq. (3) are expressed as in Eq. (4), the accumulated terms are represented as bj and cj (j= 0, 1, 2, 3, 4).

$$\left.\begin{aligned}
y_0^0 &= a_0b_0+a_1b_1+a_2b_2+a_1b_3+a_3b_4\\
y_0^1 &= a_0b_0+a_3b_1+a_1b_2+a_2b_3+a_1b_4\\
y_0^2 &= a_3c_0+a_1c_1+a_1c_2+a_1c_3+a_0c_4\\
y_0^3 &= a_0c_0+a_1c_1+a_2c_2+a_1c_3+a_3c_4
\end{aligned}\right\}\qquad(4)$$

Considering the expression in Eq. (4) the number of arithmetic operations is 20 multiplications and 16 addition operations per stage, for 2048 stages the numbers of operations are 40960 and 32768, respectively. The terms b0, b1, b2, b3, b4 and c0, c1, c2, c3, c4 are read out into the arithmetic unit twice, once for computing y_0^0 & y_0^1 and second for y_0^2 & y_0^3. To reduce the number of arithmetic operations and memory read operation, the OSA algorithm is used to compute the filter outputs.

2.1 Optimum systolic array design

To arrive at OSA algorithm, Eq. (4) is expressed as in Eq. (5), with each of the expression are computed by PE that performs two operations, multiplication of terms a_i & b_i or ai & ci and accumulation of the multiplied outputs with the previous data. The filter outputs y_0^0 & y_0^3 computation requires the filter coefficients a_i^0 = {a0, a1, a2, a1, a3} and the inputs b_i and c_i is required. The filter output y_0^1 is generated by considering the filter coefficients a_i^1 = {a0, a3, a4, a2, a1} and input data bi is required. Similarly, for generating the term y_0^2 filter coefficients a_i^2 = {a3, a1, a2, a1, a3} and input data c_i is required.

$$\left.\begin{aligned}
PE_0 &\to y_{i+1}^0 = y_0^0 + a_i^0 b_i\\
PE_1 &\to y_{i+1}^1 = y_0^1 + a_i^1 b_i\\
PE_2 &\to y_{i+1}^2 = y_0^2 + a_i^2 c_i\\
PE_3 &\to y_{i+1}^3 = y_0^3 + a_i^3 c_i
\end{aligned}\right\}\qquad(5)$$

Figure 2 presents the OSA structure designed to perform the filtering operation and the placement of PEs are carried out for optimum utilization of the filter co-

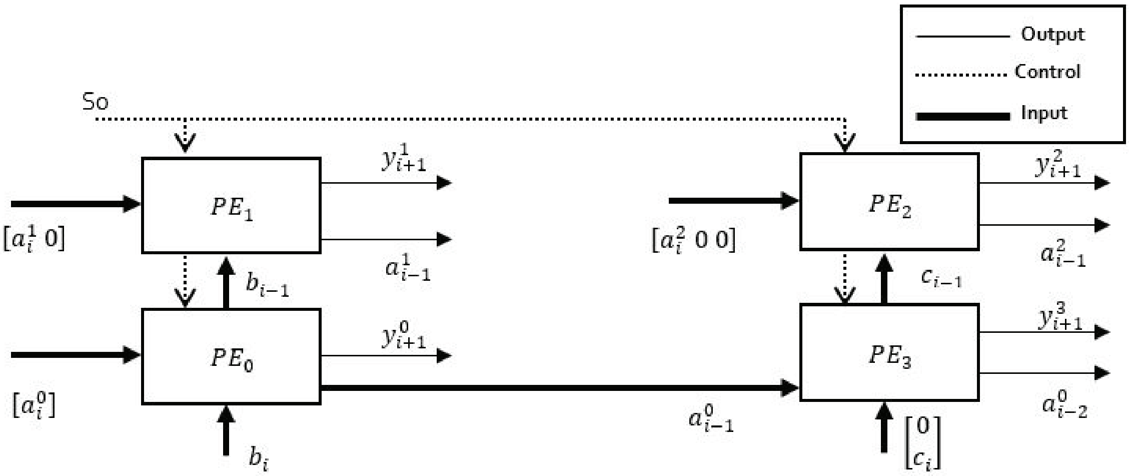


Figure 2: Optimum systolic array structure

efficients and the input data. Bold lines represent the input, the output is represented by solid lines and dotted lines represent the control input. The input data bi is orderly allowed to flow into the PE0 and then into PE1. with one clock delay, the filter coefficients a_i^0 and a_i^1 also allowed to flow into PE0 and PE1 from left to right. The filter coefficient a_i^1 is appended with '0' to synchronize with the data flow of b_i. The c_i data input is appended with '0' and a_i^2 is appended with [0 0] to synchronize with the coefficient data flow into PE3 and PE2, respectively. The internal structure of the PE is presented in Figure 3. The data inputs to the PE are u_{i-1} and v_{i-1} that flows into the PE from left and bottom. The outputs of PE are u_i, y_i and y_{i-1} that represent the data input to the next stage of PE and the output. The control signal S_0 directs the output of the demultiplexer to the output pin of PE or into the accumulator logic. The Delay Register (DR) is used to store the data ui and y_i for one clock cycle and transfers to the next PE. The register also stores the intermediate data for accumulation during the next clock cycle.

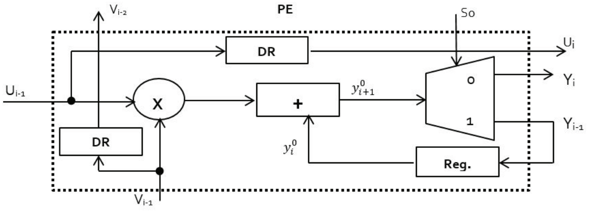


Figure 3: Internal structure of PE.

Table 2 presents the data flow and the output generation of the OSA structure shown in Figure 2 for first 12 clock cycles. The input coefficients enter from the Left (L) and the data input enter from the Bottom (B) of the PE. The first output is generated at 5th clock for the first filter, the second and third filter generates output at the 6th clock cycle and the fourth filter generates the output at the 7th clock cycle. The latency is 5 clock cycle, and the throughput is 4 outputs per clock. The advantage of the OSA structure is optimum utilization of and filter coefficients for output computation. The OSA structure designed is used in the realization of 1 to 2047 stage of the IDTCWT and 2 to 2048 stage of DTCWT. For realizing the 1st stage of DTCWT and 2048th stage of IDTCWT, MDA logic is used. Approximating the 2nd stage (Ins2) filter coefficients (±38 to ±44, ±49 to ±44and ±7 to ±6) the number of arithmetic operations is reduced.

2.2 Modified distributive arithmetic

Figure 4 presents the forward DTCWT and IDTCWT structure that performs demodulation and modulation of the OFDM signal. The corresponding filters for DTCWT and IDTCWT are represented as Is1 and Ins1 respectively and it is presented in Table 1. The four output that will be generated by the DTCWT filters is represented as $\{y_i^{00},\ y_i^{01},\ y_i^{20},\ y_i^{21}\}$ and expressed as in Eq. 6(a),(b),(c),& (d). The filter coefficients that are common are grouped together to reduce the number of multiplication operation, the corresponding input data is added prior to multiplication by the filter coefficient. From the expressions in Eq. 6 it is observed that the grouping of common terms has reduced the num-ber of multiplications to 4. The summed terms 'Y' is not common with all the four filter expressions and the filter coefficients $\{h_{00}^{'},\ g_{00}^{'}\}$ are zero which will further reduce the number of multiplication operations.

Table 2: Data flow activity in the OSA structure.

PEs/Clock		1	2	3	4	5	6	7	8	9	10	11	12
PE_0	L	a_0	a_1	a_2	a_1	a_3	a_0	a_1	a_2	a_1	a_3	a_0	a_1
	B	b_0	b_1	b_2	b_3	b_4	b_0	b_1	b_2	b_3	b_4	b_0	b_1
PE_1	L	0	a_0	a_3	a_1	a_2	a_1	0	a_0	a_3	a_1	a_2	a_1
	B	0	b_0	b_1	b_2	b_3	b_4	0	b_0	b_1	b_2	b_3	b_4
PE_3	L	0	a_0	a_1	a_2	a_1	a_3	0	a_0	a_1	a_2	a_1	a_3
	B	0	c_0	c_1	c_2	c_3	c_4	0	c_0	c_1	c_2	c_3	c_4
PE_2	L	0	0	a_3	a_1	a_2	a_1	a_0	0	0	a_3	a_1	a_2
	B	0	0	c_0	c_1	c_2	c_3	c_4	0	0	c_0	c_1	c_2
$PE_{0(Out)}$		-	-	-	-	y_0^0	y_1^0	y_2^0	y_3^0	y_4^0	y_5^0	y_6^0	y_7^0
$PE_{1(Out)}$		-	-	-	-	-	y_0^1	y_1^1	y_2^1	y_3^1	y_4^1	y_5^1	y_6^1
$PE_{3(Out)}$		-	-	-	-	-	y_0^3	y_1^3	y_2^3	y_3^3	y_4^3	y_5^3	y_5^3
$PE_{2(Out)}$		-	-	-	-	-	-	y_0^2	y_1^2	y_2^2	y_3^2	y_4^2	y_5^2

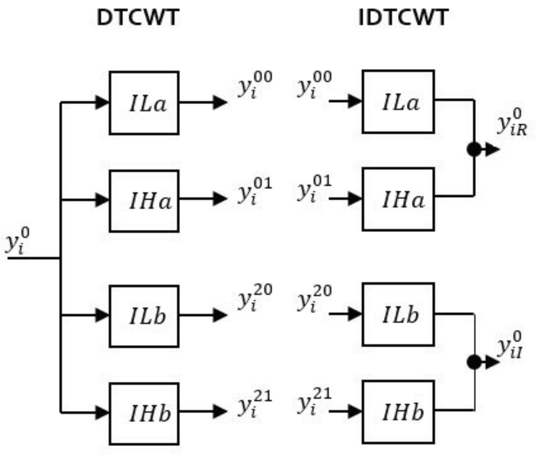


Figure 4: DTCWT and IDTCWT filters for OFDM.

The input data terms are summed up using two stage adder structures that are stored in an intermediate register. The intermediate register data is used as the address (AD) to the Look-Up Table (LUT) that is stored with precalculated partial products (PP) based on the MDA algorithm. The two stage adder structure is designed such that the register contents are used to compute two filter outputs simultaneously. Figure 5 presents the MDA structure for computing two real filters outputs y_i^{00} and y_i^{01}. The input data is loaded into the register array that loads 10 data samples. The two stage adder array performs addition operation of the data and the summed up data is stored in an intermediate register. The three multiplexers allow corresponding data into the LUT and controlled by the clock signal.

$$y_i^{00} = \begin{pmatrix} (y_0^0 + y_9^0)h_{00}^{'} + \\ \left(y_1^0 + y_2^0\right)h_{01}^{'} + \\ \left(y_3^0 + y_4^0 + y_7^0 + y_8^0\right)h_{02}^{'} + \\ \left(y_5^0 + y_6^0\right)h_{03}^{'} \end{pmatrix} \tag{6a}$$

$$y_i^{01} = \begin{pmatrix} (y_0^0 + y_9^0)h_{00}^{'} + \\ \left(y_3^0 + y_4^0)h_{03}^{'} +\right) \\ \left(y_1^0 + y_2^0 + y_5^0 + y_6^0\right)h_{02}^{'} + \\ \left(y_7^0 + y_8^0\right)h_{01}^{'} \end{pmatrix} \tag{6b}$$

$$y_i^{20} = \begin{pmatrix} (y_0^2 + y_1^2)g_{00}^{'} + \\ (y_8^2 + y_9^2)g_{01}^{'} + \\ \left(y_2^2 + y_3^2 + y_6^2 + y_7^2\right)g_{02}^{'} + \\ \left(y_4^2 + y_5^2\right)g_{03}^{'} \end{pmatrix} \tag{6c}$$

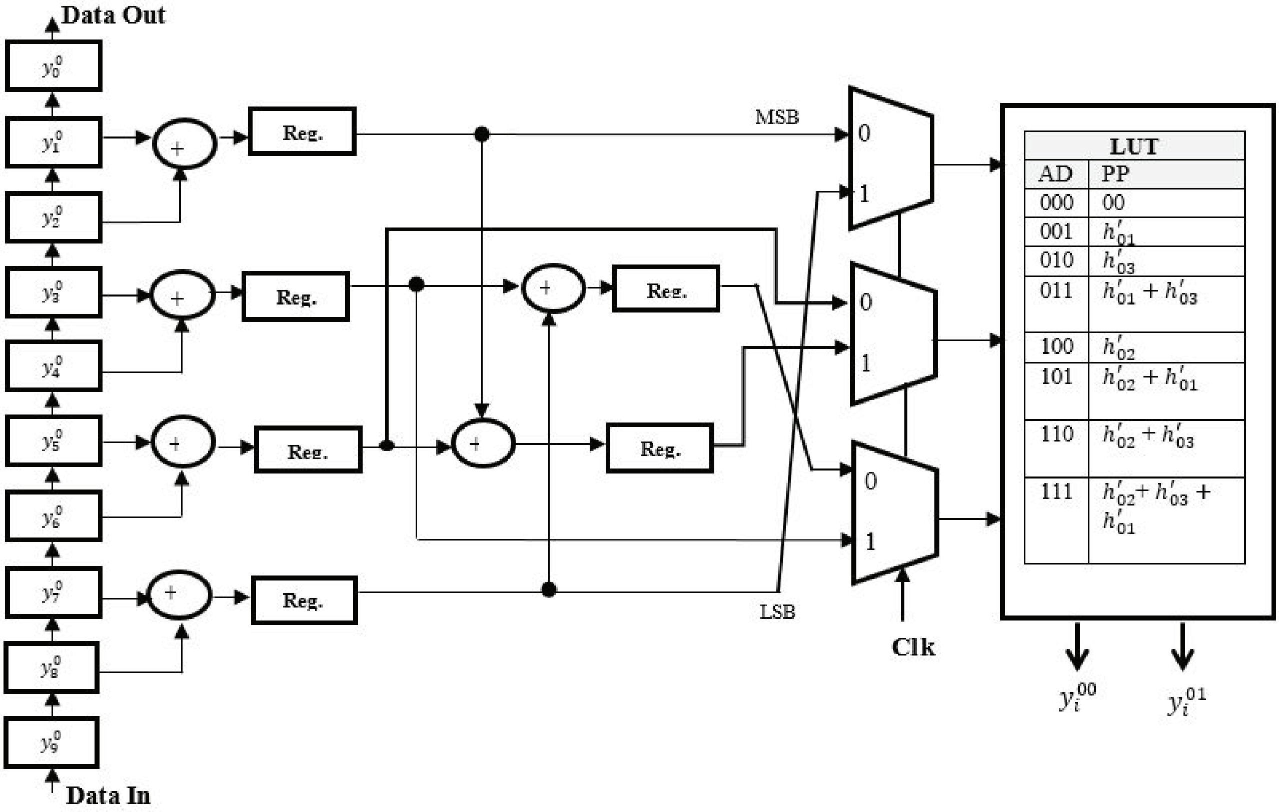


Figure 5: MDA structure for computing two real filter.

$$y_i^{21} = \left(\begin{array}{c} (y_0^2 + y_1^2) g_{01}' + \\ \left(y_2^2 + y_3^2 + y_6^2 + y_7^2\right) g_{02}' + \\ \left(y_4^2 + y_5^2\right) g_{03}' + \\ \left(y_8^2 + y_9^2\right) g_{00}' \end{array} \right) \tag{6d}$$

In the positive clock, the LUT locations are accessed to compute the first filter output y_i^{00} and in the negative clock, the PP are accessed to compute the second filter output y_i^{01}. Figure 6 presents the MDA structure for computing two imaginary filter outputs y_i^{20} and y_i^{21} The MDA logic is accessed at both positive and negative clock to compute the PP from the LUT and compute the two filter outputs. The advantage of MDA structure are that the LUT size is limited to a depth of 8 locations and used twice for computing two filter outputs. The intermediate registers are used to ensure pipelined structure and improves throughput. The first output of the filters is generated after 13 clock cycle and throughput is 4 for every clock.

2.3 Comparison of resources

The OSA structure and MDA structure designed are optimum in terms of a number of arithmetic operations and hence it is suitable for performing 2048 symbol OFDM modulation and demodulation. Table 3 compares the hardware resources and performances of the designed structure with direct implementation structure. From the comparisons of a number of multipliers and adders, the MDA structure requires 50% of LUT and the LUT depth is reduced by 99.21%. The OSA structure requires 50% of multipliers and 55% of adders. The latency is reduced by 50% due to reuse of the input data and filter coefficients. Considering the DTCWT structure designed using both OSA and the MDA algorithm, it is required to implement OFDM for subcarrier modu-

Table 3: Performance comparison.

Parameters	Direct Implementation	OSA	MDA
Filter type - DTCWT/Inverse DTCWT	Real and Imaginary	Real and Imaginary	Real and Imaginary
Filter order	10-tap	10-tap	10-tap
Multiplications	40	20	Multiplier less
Additions	36	16	14
LUTs	4	-	2
LUT depth	1024	-	8
Throughput (Output/clock)	1	4	4
Latency (clocks)	10	5	13

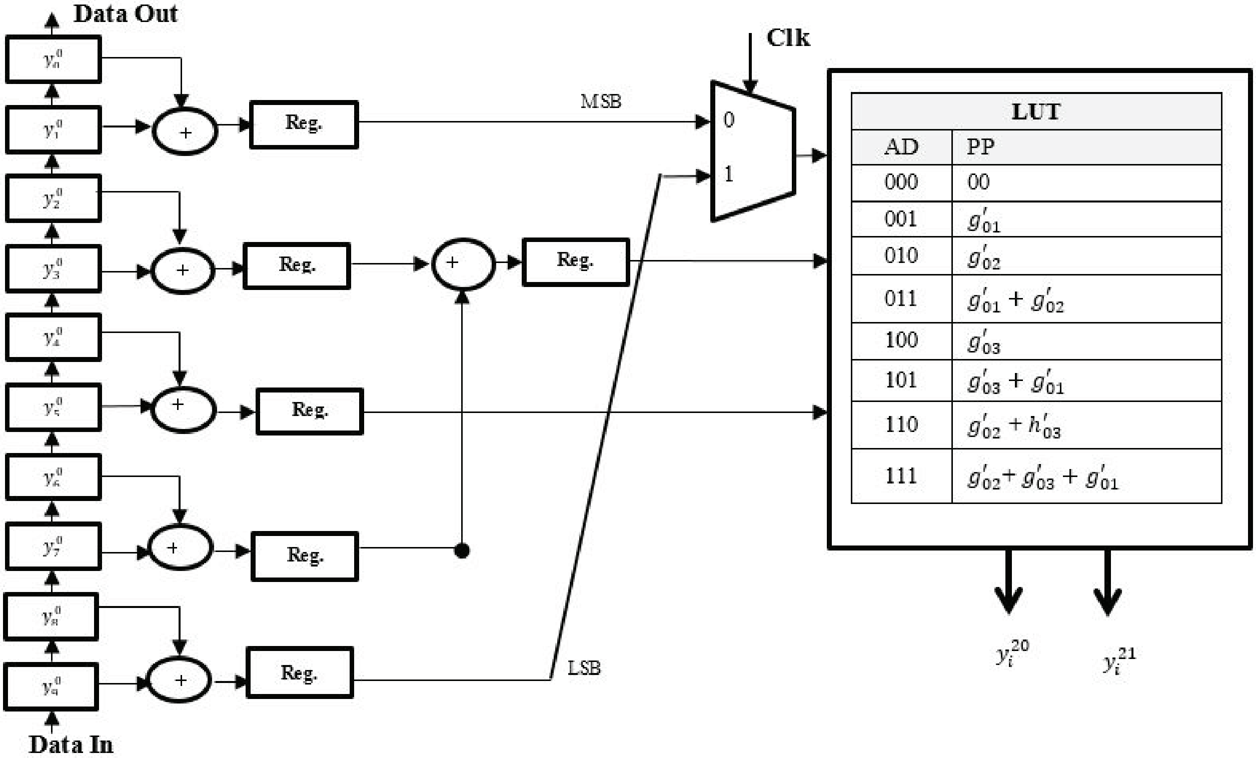


Figure 6: MDA structure for computing two Imaginary filter.

lation and demodulation. In this work, the design of 2560 subcarrier OFDM is presented.

3 DCWT OFDM using 2560 subcar-rier

According to generic underwater MODEM standards, Table. 4 presented by Kochanska et al. [20] shows the basic parameters for OFDM recommended for underwater acoustic channels for shallow water. OFDM based underwater MODEM with rank 8 and 30 kHz carrier frequency for synchronization by considering sampling rate of 200 kHz, bandwidth of 5 kHz and subcarrier spacing of 312.5 kHz. By considering 5 kHz bandwidth and number of subcarriers is 4, the subcarrier spacing will be 1250 Hz (5 KHz / 4). Increasing the number of subcarriers to 64 will impact the improvement in data rate, and the spacing between the subcarriers will be limited to 78.13 Hz leading to Inter Symbol Interference (ISI). An ideal spacing would be 312.5 Hz with number of subcarriers to be 16.Considering the standard specifications and based on discussions presented by Kochanska et al. [20] a 2560 subcarrier OFDM model is developed. The number of subcarriers is varied from 160 to 2560 and the subcarrier spacing is set between 1250 Hz to 78.13 Hz. As per the recommendations for OFDM based acoustic underwater communication, it is required to have configurable modulator and demodulator that can support 160 to 2560 subcarriers. To develop 2560 subcarrier OFDM based on DTCWT, it is required to have 2560 levels of IDTCWT for modulation and 2560 level DTCWT for demodulation. The DTCWT structure for 2560 level is presented in Figure 7. To have variation in subcarrier modulation, the structure presented in Figure 7 is designed with output taps from levels 160, 320, 640, 1280 and 2560. The 2560 structure can be configured to perform any of this modulation by selecting the outputs from intermediate stages. Every stage of DTCWT or IDTCWT uses four stages of filters that are either designed using OSA or MDA algorithm. By introducing intermediate registers between two stages, a pipeline structure can be designed for generating 2560 subcar-rier modulation. The 2560 stage pipelined structure generates 2559 detail filter coefficients and 1 approximate filter coefficients. The data movement between 2560 stages are controlled by control units that can synchronize data movement. Each stage requires 16 adders and 20 multipliers (considering OSA) has a latency of 5 clock cycles. For 2560 stage OFDM the number of multipliers and adders are 40960 and 51200. The latency for 2560 stage will be 12800 clocks. To design an optimum 2560 modulator with a trade-off between latency and arithmetic operation, a folded pipelined modulator is designed.

3.1 Folded Pipelined OFDM Modulator

In the folded pipelined OFDM modulator structure, each stage is reused twice or four times for subcarrier modulation and the data flow is controlled by a configurable logic. By designing reusable filter bank scheme, the number of filter banks required is reduced by 1280 or by 640 and hence it is called a folded structure. The first stage DTCWT comprises of four filters and generates four outputs. Stage 2 to stage 2560 of DTCWT comprises of two filters that are grouped together and process the data from preceding stage to generate two outputs from every pair or group as shown in Figure 7. The OSA structure shown in Figure 2 is designed for processing data input and to generate four outputs (real and imaginary). The reduced optimum OSA structure is designed to generate two outputs as shown in Figure 8 (a) for the real part and (b) for imaginary part separately.

Figure 9 presents the top-level block diagram of proposed folded pipelined unit for computing 2560 subcarrier OFDM demodulation using forward DTCWT. The first stage is realized using the MDA algorithm and generates four outputs. One output from each pair of filters $\{ y_i^{00}, y_i^{20} \}$ is considered for the next level of processing. The folded pipelined structure consists of N stage of processing units (N=1280) and each processing unit has two Fold Units (FLU) and two compute unit. The compute unit is the OSA shown in Figure 8. The FLU is designed to realize either two or four stages of DTCWT decomposition. By reusing the processing units twice, two-stage decomposition is carried out and by reusing

Table 4: OFDM parameters for underwater acoustic communication.

No. of Subcarriers NS	No. of Subcarriers NB in B=5 KHz	Subcarrier Spacing B_S [Hz]	Symbol Duration T_{OFDM} [ms]	Symbol Duration with CP T_S [ms]	Symbols Per Frame
160	4	1250.0	0.80	1.00	2500
320	8	625.0	1.60	2.00	1250
640	16	312.5	3.21	4.01	625
1280	32	156.3	6.41	8.01	312
2560	64	78.13	12.82	16.03	156

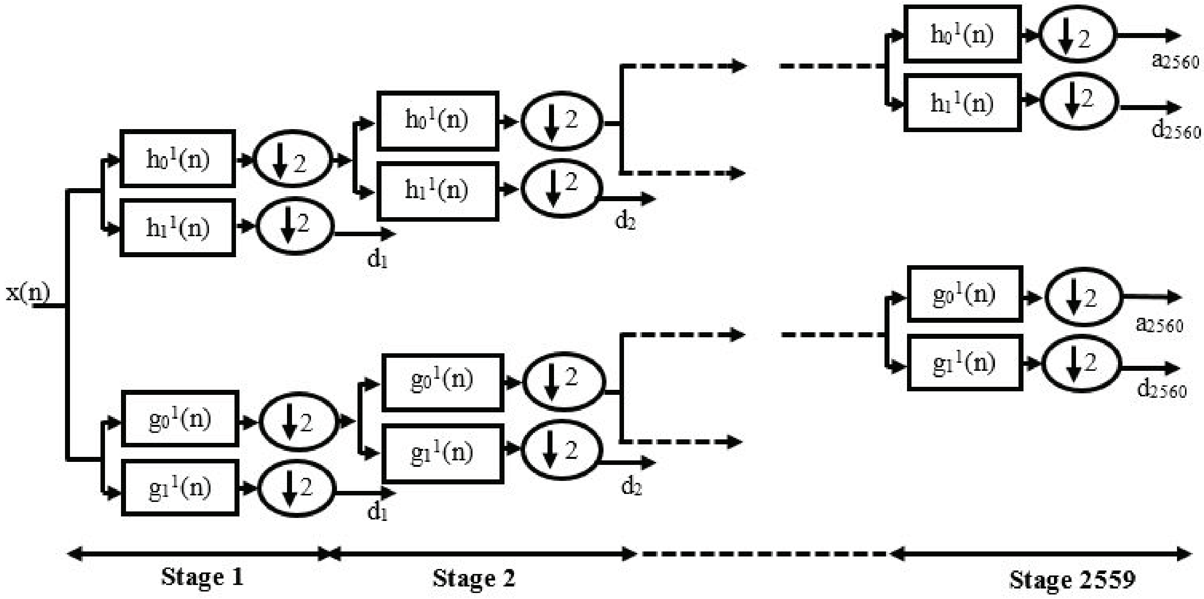


Figure 7: DTCWT structure for 2560 stages.

four times, four stages of DTCWT are achieved. Stage 0 in Figure 9 is implemented using the proposed MDA logic discussed in section 2.2 and stage 1 to stage N is realized using the OSA structure discussed in section 2.1. Figure 10 presents the fold unit design that comprises of input register of depth 18 that stores the input data from the previous stages denoted as x. The register contents of each memory location of input register is accessed and denoted as R. The outputs of input register are connected to the multiplexer array so that the data R0 to R18 are rearranged as shown at the inputs of the multiplexer array. Select line S1 and S0 are used to configure the fold unit to either fold by two or fold by four. If the select signal S0 is alone used then fold by two logic is achieved and if both S0 and S1 are used, then fold by four logic is achieved. The data enters the OSA structure for processing and generates two outputs of which one of the outputs y_{i+1}^{0} is de-multiplexed and stored in the output register (Q) for next stage processing. The other data y_{i+1}^{1} is demodulated subcarier forwarded to last stage of OFDM demodulator. The demultiplexer logic and input stage multiplexer array are synchronized for computing either two levels or four levels DTCWT decomposition. The de-multiplexer output denoted as {Q0, Q1, Q2, Q3} are stored into a corresponding memory location in the output register for reuse and for computing the next level decomposition. The read and write signal are used to read out the data into the multiplexer array form the input register and write the output of OSA into the output register. After every read and write operation is performed the input and output registers are shifted to load new data inputs for processing. Figure 11 and Figure 12 presents PE configured to perform decomposition by two or four. In Figure 11, the input stage for fold by two logic consists of two data array registers of depth four, represented as 'x' and 'y'. The input register array 'x' is loaded with new input data at the register (xi + 4). At every clock, the data is shifted up in the data register 'x'. The output of OSA is demultiplexed and shifted as input to register 'y' and again the cycle repeats. In the first clock the data input 'x' is processed by OSA to generate two outputs y_i^{00} and y_i^{01} of which y_i^{00} is shifted back for a second level of processing into the 'y' register.

In the next clock pulse, the data from 'y' register is multiplexed and processed by OSA logic, which generates two inputs, of this one of them is de-multiplexed and sent to next stage for processing. Similarly, Figure 12 presents the logic of fold by four that comperes of four input array registers represented as {x, y, z, w}, each of depth four. The fold by four module is designed to pro-

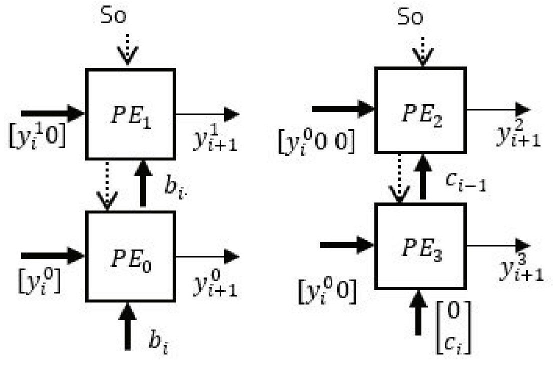


Figure 8: Optimum systolic array (a) Real part (b) Imaginary part.

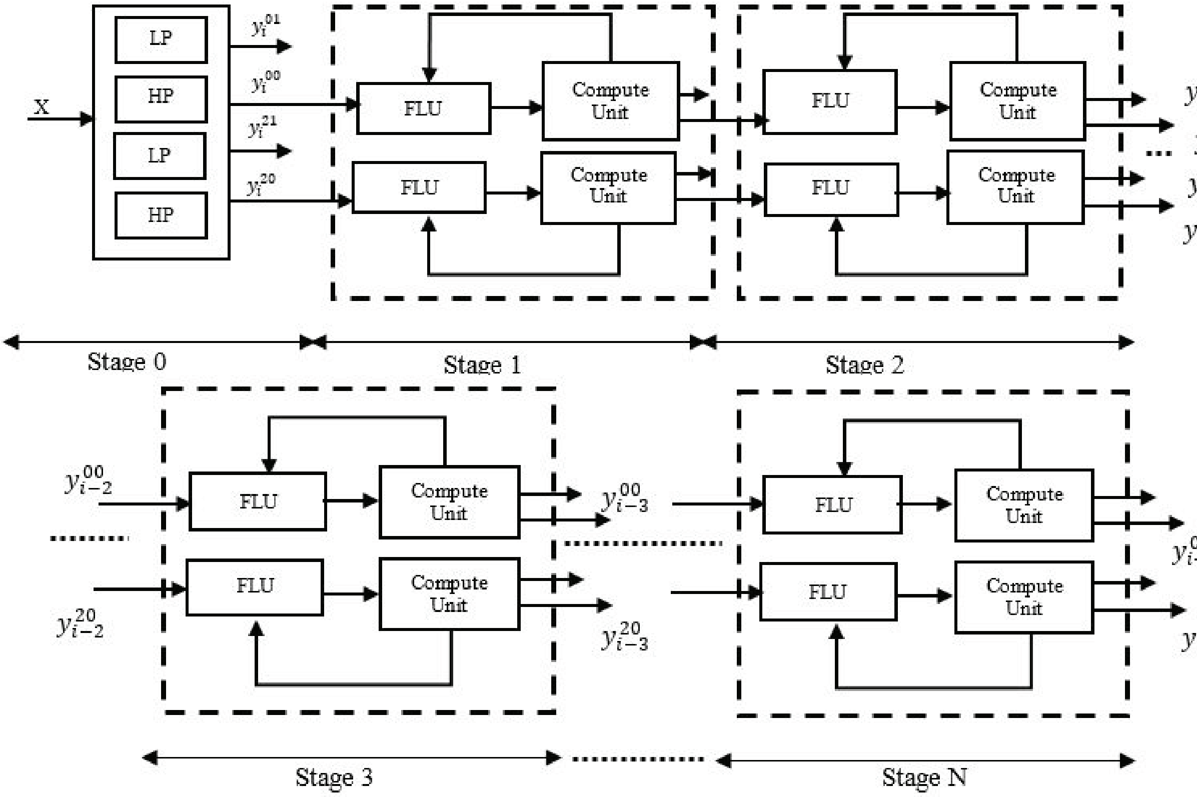
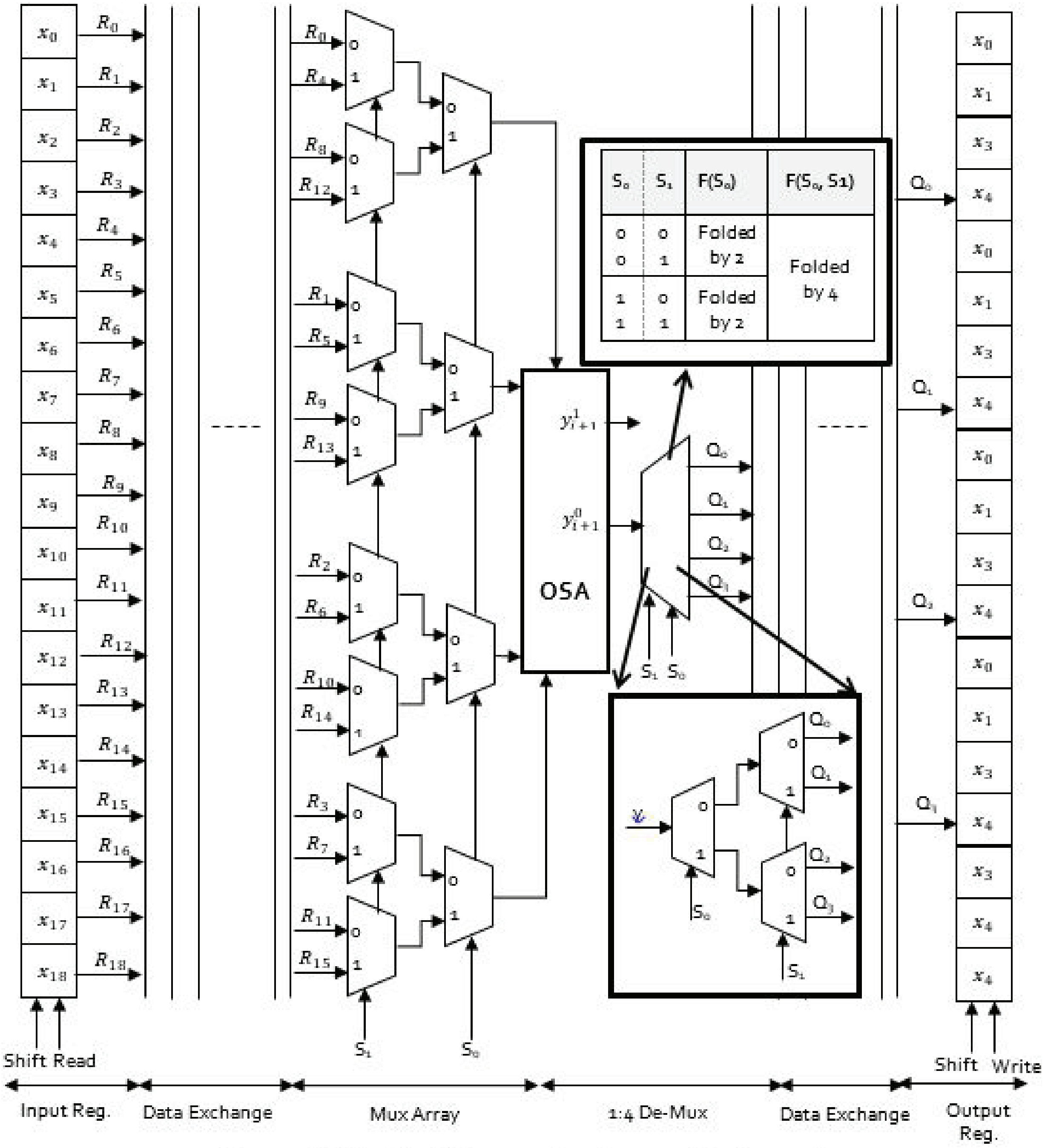


Figure 9: Folded pipelined OFDM demodulator block diagram.

cess subcarrier modulation by reusing DTCWT filter pair four times. The control signal S0 and S1 are set to perform fold by two or fold by four operations. The data flow logic designed for fold by two logic is presented in Figure 13. There are two registers at input represented as 'y' and 'x'. In the first clock, the four inputs x0, x1, x2 and x3 are loaded into 'x' register array whereas 'y' array is set to zero. The OSA unit generates y0 at 2nd clock cycle that gets loaded into the fourth register of 'y'. In every clock, the input 'x' is processed to generate output 'y' and will be loaded into the 'y' array register. Once all the contents of 'y' register are computed (clock 5), the multiplexer at the input of OSA is enabled to process 'y' data. The two-stage fold logic generates output for alternate stages i.e stage 1 to stage 3. Similarly, Figure 14 presents a data flow diagram for fold by four logic. In this logic, the input data x0, x1, x2 and x3 are loaded in a first clock cycle and the first output of the 1st stage is computed at second clock pulse. The 1st output of the third stage (z0) is computed at 11th clock and the 1st output of 4th stage (w0) is computed at 16th clock. The fold by four logic processes data from the 1st stage to generate data to the 5th stage. The folded pipelined architecture is configured to compute N=2560 subcarrier OFDM symbols using only N/2 stages in the fold by two logic and N/4 stages using fold by four logic. In fold by two logic, the latency is 10 clock cycles for every six outputs and in the fold by four logic, the latency is 20 clocks for every ten outputs. The number of multiplier and adder operations are reduced by 50% in fold by two as compared with direct implementation and reduced by 75% in fold by four logic. A trade-off between computation complexity and latency is achieved in the folded pipelined architecture. This structure can be configured to compute 160, 320, 640, 1280 and 2560 subcarrier modulation by taping the outputs at N/2, N/4, N/8 and N/16 stages. The 2560 stage DTCWT decomposition unit is modelled using Verilog HDL and is verified for its functionality. The functionally correct HDL code for DTCWT and IDTCWT is implemented on FPGA and the logic correctness of OFDM module is verified in system generator environment.

4 FPGA Implementation

Figure 15 presents the top-level block diagram of IDTCWT-DTCWT validation model. Input signal represented as in Eq. (7) is generated in the Matlab Simulink environment. Two frequencies of 40 kHz and 100 kHz are used that are quantized to 8-bit numbers as a composite input signal. The parameters q1 and q2 are quantization factors to scale the input to nearest integer numbers. Each of the samples of composite input


x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15} x_{16} x_{17} x_{18}
R_0 R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11} R_{12} R_{13} R_{14} R_{15} R_{16} R_{17} R_{18}
0
1
y^1_{i+1}
y^0_{i+1}
OSA
Q_0 Q_1 Q_2 Q_3
S_1 S_0
y_i
Shift Read
Shift Write
Input Reg.
Data Exchange
Mux Array
1:4 De-Mux
Data Exchange
Output Reg.

S_0	S_1	$F(S_0)$	$F(S_0, S_1)$
0 0	0 1	Folded by 2	Folded by 4
1 1	0 1	Folded by 2	

Figure 10: Fold unit logic for computing decomposition by 2 and 4.

data is encoded using 64-QAM modulation scheme in the Simulink environment.

Input=q1*round ((10sin (2*pi*f1*t) +10)/q1+q2)
*round ((10sin (2*pi*f2*t) +10)/q2) (7)

The QAM data from Simulink environment is read into system generator model through gateway in port. The symbols from the gateway in port are converted to parallel data and are processed by the IDTCWT unit that is modelled using Verilog code. The four filter outputs of IDTCWT module which is a complex data is converted to real data and is further processed by the DTCWT model. OFDM modulation and demodulation is per-formed by the IDTCWT-DTCWT pair. The output of DTCWT is read into the Simulink environment through gateway out module. From the results obtained in the workspace of MATLAB the demodulated data symbols are processed by the inverse QAM module to generate the output signal. Figure 16 presents the simulation results of the system generated module for DTCWT stage 1 demodulator. As the input from Simulink envi-ronment is quan-

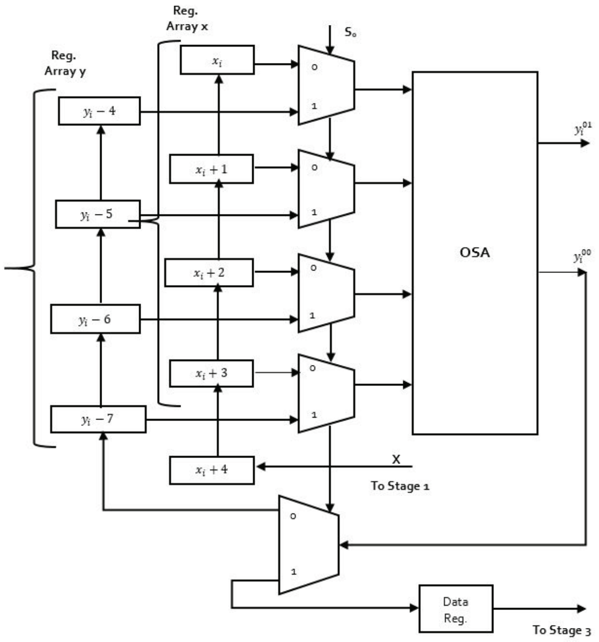


Figure 11: Computing two stage decomposition using one stage DTCWT.

tized to positive integers, the sine wave generated is used as an input sequence for OFDM modulation. The modulated data is processed by the DTCWT module and OFDM demodulation is carried out. The input sequence and the demodulated output sequence is seen and verified for its numerical values. At the output, the numerical values obtained are equal to the input samples but a delay of 3 clocks is seen.

Figure 17 presents the FPGA implementation of OFDM modulation and demodulation on the Virtex-5 development kit. The input data from Simulink environment is used as the source to the system generator model and the same input is provided to the FPGA device for performing modulation and demodulation process. The output of FPGA is read back using chip scope debugging tool for validation of the designed model. The OFDM model implemented on FPGA is verified for its logic correctness and the hardware implementation report generated is analyzed to show area, power, and timing parameters. A detailed discussion on FPGA results is presented in the next section.

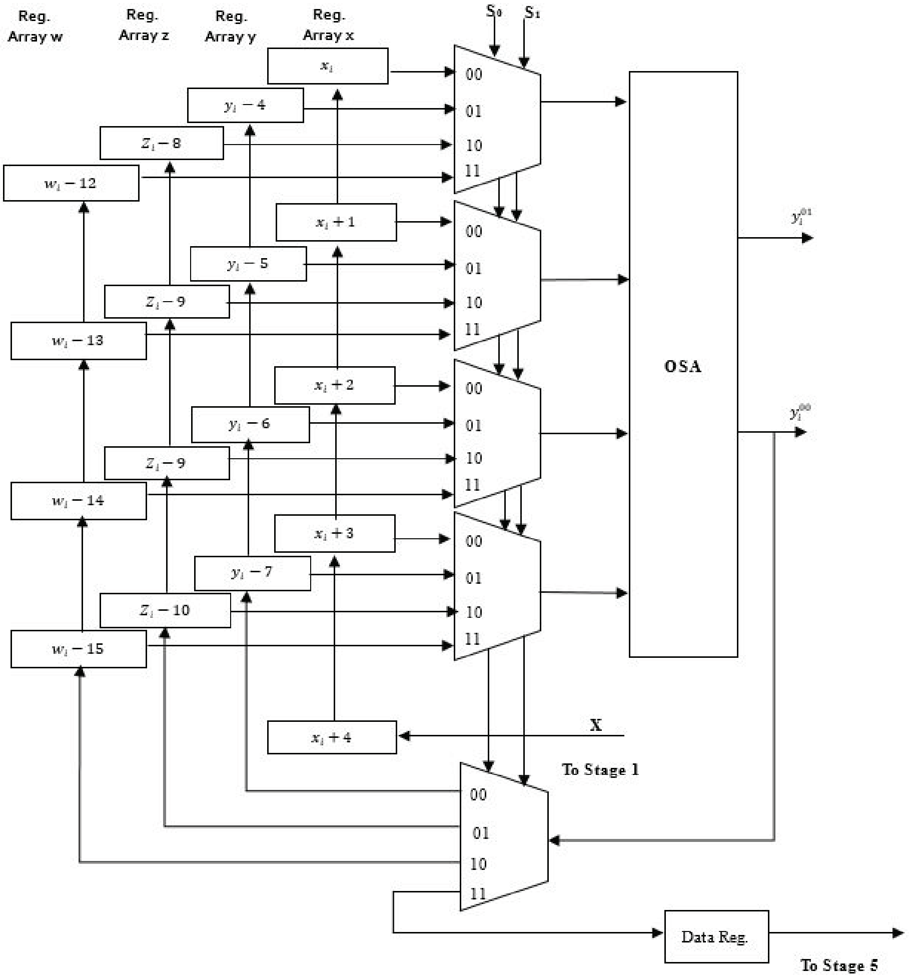


Figure 12: Computing four stage decomposition using one stage DTCWT.

Clk	1		2		3		4		5		6		7			15	
Reg.	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	----	Y	X
i	0	X_0	0	X_1	0	X_2	0	X_3	Y_0	X_4	Y_2	X_6	Y_2	X_6	----	Y_{10}	X_{14}
i+1	0	X_1	0	X_2	0	X_3	Y_0	X_4	Y_1	X_5	Y_3	X_7	Y_3	X_7	----	Y_{11}	X_{15}
i+2	0	X_2	0	X_3	Y_0	X_4	Y_1	X_5	Y_2	X_6	Y_4	X_8	Y_4	X_8	----	Y_{12}	X_{16}
i+3	0	X_3	Y_0	X_4	Y_1	X_5	Y_2	X_6	Y_3	X_7	Y_5	X_9	Y_5	X_9	----	Y_{13}	X_{17}

Figure 13: Data flow in decomposition by two.

Clk	1				2				3				4				5			
Reg.	W	Z	Y	X	W	Z	Y	X	W	Z	Y	X	W	Z	Y	X	W	Z	Y	X
i	0	0	0	X_0	0	0	0	X_1	0	0	0	X_2	0	0	0	X_3	0	0	Y_0	X_4
i+1	0	0	0	X_1	0	0	0	X_2	0	0	0	X_3	0	0	Y_0	X_4	0	0	Y_1	X_5
i+2	0	0	0	X_2	0	0	0	X_3	0	0	Y_0	X_4	0	0	Y_1	X_5	0	0	Y_2	X_6
i+3	0	0	0	X_3	0	0	Y_0	X_4	0	0	Y_1	X_5	0	0	Y_2	X_6	0	0	Y_3	X_7
Clk	9				10								14				15			
Reg.	W	Z	Y	X	W	Z	Y	X					W	Z	Y	X	W	Z	Y	X
i	0	Z_0	Y_4	X_8	0	Z_1	Y_5	X_9	----------------				W1	Z_5	Y_9	X_{13}	W_2	Z_6	Y_{10}	X_{14}
i+1	0	Z_1	Y_5	X_9	0	Z_2	Y_6	X_{10}					W2	Z_6	Y_{10}	X_{14}	W_3	Z_7	Y_{11}	X_{15}
i+2	0	Z_2	Y_6	X_{10}	0	Z_3	Y_7	X_{11}					W3	Z_7	Y_{11}	X_{15}	W_4	Z_8	Y_{12}	X_{16}
i+3	0	Z_3	Y_7	X_{11}	0	Z_4	Y_8	X_{12}					W4	Z_8	Y_{12}	X_{16}	W_5	Z_9	Y_{13}	X_{17}

Figure 14: Data flow in decomposition by four.

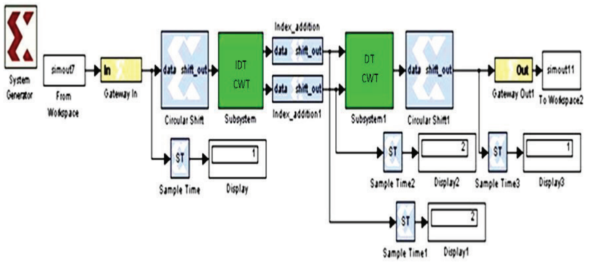


Figure 15: System generator model for validation of IDTCWT-DTCWT model.

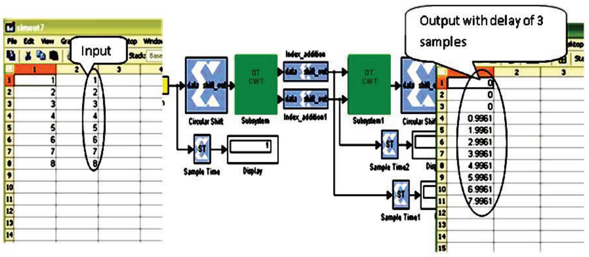


Figure 16: Validation of OFDM modulation using DTCWT.

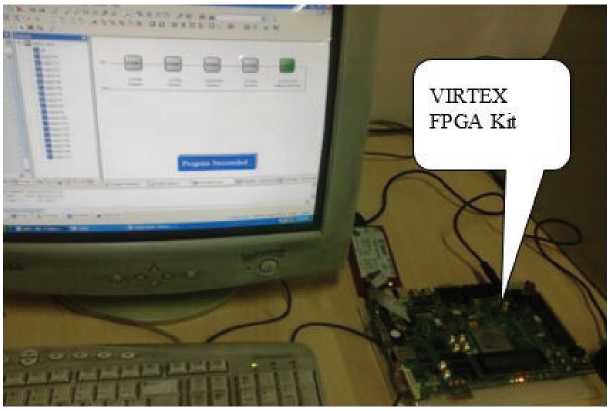


Figure 17: FPGA implementation of OFDM model.

5 Results and Discussion

The functional correct HDL model is synthesized targeting Virtex-5 FPGA family and synthesis report is obtained. Verilog HDL code is developed to model the proposed DTCWT calculation unit. FSM is designed to model control logic that synchronizes the forward transform operation. The input stage consists of serial to parallel converter realized using de-multiplexer and multiplexer that is designed to work as a parallel to serial converter at the output stage. The DTCWT module is modelled using Xilinx IP (Internet Protocol) cores and glue logic. A test bench is developed that uses known test vectors (Sinusoidal signal with center frequency 14 KHz, -3dB bandwidth of 5.6 KHz, effective bit rate 1222 b/s, with transition from 0 to 2v and each bit represented by signed integer of 8 bits) to verify logic correctness of the developed Verilog HDL model. Input data symbols that are represented using 8-bit signed representation are stored in test bench and are forced

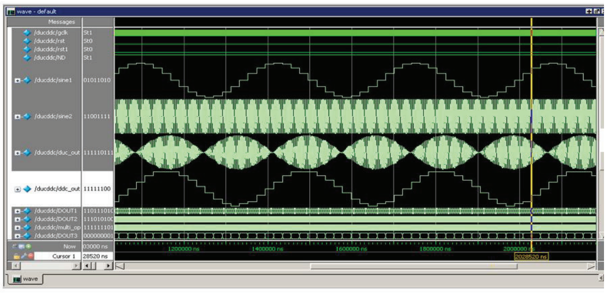
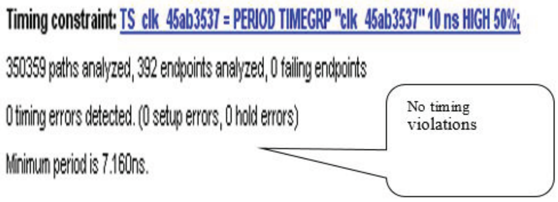


Figure 18: Simulation results of OFDM module.

into the HDL model for DTCWT calculation. The DTCWT coefficients computed by the HDL model are seen for its numerical values and compared with theoretical values. From the comparison of practical and theoretical values, the logic correctness of HDL code is verified. In addition to processing modulation symbols, the filter is verified for its impulse response and the output of the filter is seen to produce filter coefficients. The first stage four filter bank structure results are verified for impulse response and the output is seen to be the filter coefficients proving logic correctness of Verilog HDL and architecture design. The functionally verified HDL code is synthesized and RTL schematic is obtained for one filter and four filter structure. Figure 18 presents the simulation results of OFDM modulator and demodulator captured in Modelsim environment for an input sine wave. The input sine wave is modulated using OFDM modulator and the results seen at one of the four filters are presented. At the receiver, the DTCWT module demodulates the OFDM signal and the data is recovered at the output. From the simulation results, the input and output wave are matching as per the requirement. Figure 19 presents the timing report of OFDM modulator, from the timing report it is seen that for the clock period of 10ns with 50% duty cycle there is no errors found after analyzing 350359 timing paths with 392 endpoints. There is no setup and hold time violations as well and the minimum time is seen to be 7.16 ns which gives a maximum operating frequency of 139 MHz. Figure 20 presents the hardware results captured using chip scope debugging tool at the output of IDTCWT module. The peaks are seen at regular intervals and the pattern is seen to be like the results seen in Modelsim environment.



Timing constraint: TS_clk_45ab3537 = PERIOD TIMEGRP "clk_45ab3537" 10 ns HIGH 50%;

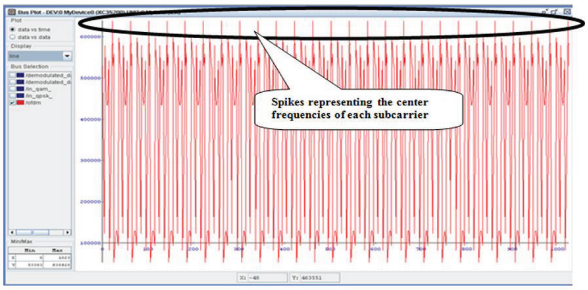
350359 paths analyzed, 392 endpoints analyzed, 0 failing endpoints

0 timing errors detected. (0 setup errors, 0 hold errors)

Minimum period is 7.160ns.

No timing violations

Figure 19: Timing report of OFDM module



Spikes representing the center frequencies of each subcarrier

Figure 20: Chip scope debugging results of OFDM modulator

Table 5 summarizes the FPGA implementation report of the DTCWT filter alone. Each of the four filters is implemented on FPGA independently and area, timing and power report is generated. Finally, DTCWT structure for 2560 stage is implemented on FPGA. The 2560 stage DTCWT filter runs at a maximum frequency of 248 MHz consuming power dissipation of less than 1.33W occupying 9982 LUTs and 32 DSP arithmetic resources. The results presented in this paper are the first set of information of implementing DTCWT on FPGA for OFDM applications. Table 6 compares the FPGA implementation results of 1D – DTCWT level structure. The proposed single-stage structure operates at a maximum frequency of 302.87 MHz with power dissipation of less than 0.82 W consuming less than 376 slices. The first stage is designed using MDA logic and hence the number of LUTs is very less as compared with all other implementations.

Table 5: Summary of Synthesis Report.

Parameter	One filter	Two filter	Four filter	2560 stage
Number of Slice Registers	87	168	376	9982
Number of Slice LUTs	88	170	373	9982
Number of fully used LUT FF pairs	81	156	353	9982
Number of bonded IOBs	24	37	68	436
Number of BUFG/ BUFGCTRLs	1	1	1	32
Number of DSP48A1s	1	2	4	32
Maximum Frequency (MHz)	489.89	374.5	302.87	248.23
Total Supply Power (W)	0.37	0.37	0.82	1.33

Table 6: Comparison of hardware requirements.

	Ref [17]	DTCWT [16]	Ref [15]	This work
Number of Slice Registers	1836	2056	3741	376
Number of Slice LUTs	1586	2045	3612	373
Total power (W)	0.7851	0.85	1.001	0.82
Maximum Frequency (MHz)	291.12	278.89	246.76	302.87

6 Conclusion

The configurable DTCWT based OFDM modulator-demodulator is designed and is implemented on FPGA. The 2560 stage DTCWT OFDM structure is configurable to perform 160, 320, 640, 1280 and 2560 level subcarrier modulation. Optimum systolic array (OSA) unit is designed with PE computing 4 outputs at every clock with latency of 5 clocks. The Modified Distributive Arithmetic (MDA) unit computes two filter outputs with throughput of four and latency of 13 clocks optimizing LUT size to 99.21%. Folded pipelined OFDM modulator is designed using fold unit logic to either perform two stage decomposition or four stage decomposition. The 2560 stage OFDM modulator is realized using folded pipelined structure operating at maximum frequency of 248 MHz consuming power less than 1.33 W. With low power and high processing speed, the OFDM structure is suitable for underwater communications that requires adaptive modulation scheme.

7 Patent

Application no. - 202041023845
Date of filing the application: 06-06-2020
Publication Date: 12-02-2021
Title of Invention: Method for Performing OFDM Modulation and Demodulation Based on DTCWT with N-Subcarriers Designed using Pipelined-Folded-Reusable Systolic Array Algorithm with Reconfigurable Logic.
Name of Inventors: Girish N, Veena M B, Cyril Prasanna Raj

8 Conflict of Interest

We hereby declare that there is no conflict of interest in publishing this paper.

9 References

1. A. Goalie, J. Trubuil, N. Beuzelin,"Channel coding for underwater acoustic communication sys-tem," IEEE Oceans 2006, September 18-21, Bos-ton, MA, USA, 2006. https://doi.org/10.1109/OCEANS.2006.307093
2. Po-Cheng Wu, Liang-Gee Chen,"An efficient architecture for two-dimensional discrete wavelet transform," IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 4, pp. 536–545, 2001. https://doi.org/10.1109/VTSA.1999.786013
3. Wim Sweldens,"The lifting scheme: A custom-design construction of biorthogonal wavelets," Journal of Applied and Computational Harmonics Analysis, vol. 3, no. 2, pp. 186-200, 1996. https://doi.org/10.1006/acha.1996.0015
4. A.S. Lewis, G. Knowles,"VLSI architecture for 2-D daubechies wavelet transform without multipliers," Electronics Letters, vol. 27, no. 2, pp. 171–173, 1991. https://doi.org/10.1049/el:19910110
5. C. Chakrabarti, M. Vishwanath,"Efficient realizations of the discrete and continuous wavelet transform: from single chip implementations to mapping on SIMD array computers," IEEE Trans-actions on Signal Processing, vol. 43, no. 3, pp.759–771, 1995. https://doi.org/10.1109/78.370630
6. M.B.Veena, M.N.Shanmuka Swa-my,"Performance analysis of DWT based OFDM over FFT based OFDM and implementing on FPGA," International Journal of VLSI design and Communication Systems (VLSICS), vol.2, no.3, September, 2011. http://doi.org/10.5121/vlsic.2011.2310
7. A.Grzeszczak, M.K.Mandal, S.Panchana-than,"VLSI implementation of discrete wavelet transform," IEEE Transactions on VLSI Systems, vol.4,no.4,1996. https://doi.org/10.1109/92.544407
8. Chao Cheng, Keshab K. Parhi,"High Speed VLSI Implementation of 2-D Discrete Wavelet Transform," IEEE Transactions on Signal Processing, vol. 56, no.1, pp. 393-403, 2008. https://doi.org/10.1109/TSP.2007.900754
9. S. Masud, J. V. McCanny,"Reusable Silicon IP Cores for Discrete Wavelet Transform Applica-tions," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 51, no. 6, pp. 1114-1124, 2004. https://doi.org/10.1109/TCSI.2004.829236
10. Chengjun Zhang, ChunyanWangand, M. Omair Ahmad,"A Pipeline VLSI Architecture for Fast Computation of the 2-D Discrete Wavelet Trans-form," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 59, no. 8, 2012. https://doi.org/10.1109/TCSI.2011.2180432

11. XinTian, Lin Wu, Yi-Hua Tan, Jin-Wen Tian,"Efficient Multi-input/Multi-output VLSI ar-chitecture for 2-D lifting-based discrete wavelet transform," IEEE Transaction on Computers, vol. 60, no. 8, pp. 1207–1211, 2011. https://doi.org/10.1109/TC.2010.178
12. Yeong-Kang Lai, Lien Fei Chen, Yui-Chih Shih,"A high-performance and memory efficient VLSI ar-chitecture with parallel scanning method for 2-D lifting based discrete wavelet transform," IEEE Transaction on Consumer Electronics, vol. 55, no. 2, pp. 400-407, 2009. https://doi.org/10.1109/TCE.2009.5174400
13. Chih-Chi Cheng, Chao-Tsung Huang, Ching.-Yeh Chen, Chung-JrLian, Liang-Gee Chen,"On-chip memory optimization scheme for VLSI implementation of line-based two-dimensional discrete wavelet transform," IEEE Transaction on cir-cuits and System for Video Technology, vol. 17, no. 7, pp. 814-822, 2007. https://doi.org/10.1109/TCSVT.2007.897106
14. S. S. Divakara, Sudarshan Patilkulkarni, Cyril Prasanna Raj,"High Speed Area Optimized Hybrid DA Architecture for 2D-DTCWT," International Journal of Image and Graphics, vol. 18, no. 01, 2018. https://doi.org/10.1142/S0219467818500043
15. S. S. Divakara, Sudarshan Patilkulkarni, Cyril Prasanna Raj,"High Speed Modular Systolic array based DTCWT with Parallel Processing Architecture for 2D Image Transformation on FPGA," International Journal of Wavelets, Multiresolution and Information Processing, vol. 15, no. 5, 2017. https://doi.org/10.1142/S0219691317500473
16. G. Venkateshappa, Cyril Prasanna Raj,"Design of DTCWT-DWT Image Compressor- Decompressor with commanding Algorithm", European journal of Advances in Image and Video Processing, vol.5, no. 1, 2017. https://doi.org/10.14738/aivp.51.2777
17. B. Poornima, A. Sumathi, Cyril Prasanna Raj,"Memory efficient high speed systolic array architecture design with multiplexed distributive arithmetic for 2D DTCWT computation on FPGA," journal of microelectronics, electronic components and materials, vol. 49, no. 3, 2019 https://doi.org/10.33180/InfMIDEM2019.301
18. H. T. Kung, Charles E. Leiserson,"Algorithms for VLSI processor arrays," Introduction to VLSI Systems, pp. 271-292, 1980. http://www.eecs.harvard.edu/~htk/publication/1980-introduction-to-vlsi-systems-kung-leiserson.pdf
19. M. Y. Chern, T. Murata," Efficient matrix multiplications on a concurrent data-loading array processor," IEEE ICPP, pp. 90-94, 1983. https://www.osti.gov/biblio/5364346
20. Iwona Kochańska, Jan H. Schmidt, Jacek Marszal,"Shallow Water Experiment of OFDM Underwater Acoustic Communications," Archives of Acoustics, vol. 45, no. 1, pp. 11–18, 2020. https://doi.org/10.24425/aoa.2019.129737

Arrived: 14.12.2020
Accepted: 13.05.2021