AN APPARATUS FOR FAST IMAGE REGISTRATION ALGORITHM WITH DTCWT COMPUTATION USING FOUR STAGE PIPELINED FILTER BANK ARCHITECTURE INTEGRATED WITH HYBRID DISTRIBUTIVE ARITHMETIC ARCHITECTURE SUSBSYSTEM

PREAMBLE OF THE DESCRIPTION:

THE FOLLOWING SPECIFICATION PARTICULARLY DESCRIBES THE INVENTION AND THE MANNER IN WHICH IT IS TO BE PERFORMED

A) TECHNICAL FIELD OF INVENTION

[001] The present invention relates to architectures for implementing Four Stage Pipelined Architecture for Discrete Time Complex Wavelet Transforms (DTCWTs) on Field Programmable Gate Array Platform. The invention relates to image registration applications where DTCWTs may be used for computing sub bands for a given input signal. The present invention relates to computing DTCWT using distributive arithmetic algorithm and multiplexers.

B) BACKGROUND OF THE INVENTION

[002] Image registration is one of the most important image processing techniques for extraction of information from input data acquired from different image sensing sources. Two images captured from different sources depicting the same source are registered in image registration process [1]. Image sensors capture objects in the field of view that are of different shapes, dimensions, colour, opaqueness, surface texture and location. The objects appearing in the images captured provide information to the user that can lead to several applications such as remote sensing, surveillance, medical diagnosis etc. The most frequently arising problems in the processing of (still) images are that of object registration. It arises in images containing objects, possibly overlapping, against a more-or-less uniform background. Challenges in image registration are to detect the features of objects from the images captured and generate registered image that matches with the reference image [2]. Image registration is carried out by performing feature extraction, matching of features, estimating transformation parameters and resampling. Feature based

algorithm are faster and are preferred in high resolution image registrations as compared with pixel based image registration algorithms. In this work, image registration algorithms based on wavelet features are designed demonstrating improvement in image registration accuracy and fast algorithms are developed for real time applications.

[003] In wavelet based image registration technique, the images to be registered are first transformed into multi resolution sub bands. Sub bands are combined using various fusion rules such as maximum selection scheme, weighted average scheme, window based verification scheme, activity based measurement and coefficient grouping method [5, 6]. The number of levels for DWT decomposition is recommended to be 6 levels for bior1.3 wavelet filter [6, 7]. Computing 6 level DWT for both the image to be fused is a complex process as they require large number of arithmetic operations to be performed and the operations also depend upon number of wavelet filter coefficients.

[004] However, there are certain limitations in wavelets such as time variant and directionality that are addressed by use of dual tree wavelet transforms [8]. If the image to be fused have shifts as compared with reference images wavelets exhibit shift variance property thus altering the energy coefficients across the wavelet sub bands. Fusion of images needs to account for this property and hence use of DTCWT is recommended for image fusion. DCTCWT is also been used for image registration also. Introduction of the DTCWT was made in [9], and showed which has desirable properties of approximate shift insensitive, good directionality, perfect reconstruction with use of short linear phase filters, limited redundancy and efficient order-N computation.

[005] DTCWT uses real DWT and imaginary DWT to decompose a signal in terms of complex shifted and dilated mother wavelets and scaling functions. For the 1D decomposition, DTCWT requires four filters (two pairs of low pass and high pass) for real and imaginary coefficients computation. For 2D DTCWT computation along with four filters, eight additional filters are required that produces six sub bands along with two low pass sub bands [10]. For fusion of images using DTCWT, six levels decomposition is required, and for each level twelve filter banks are required. Thus the computation complexity of DTCWT is higher than computation complexity in DWT.

[006] Complexity in terms of speed, area and power dissipation of DTCWT and IDTCWT architectures are the design challenges for VLSI engineers. There are two most popular schemes for computation of DTCWT, lifting algorithm and distributive arithmetic algorithm. In lifting algorithm the input data is processed by predict and update modules to compute the low pass and high pass filter coefficients based on suitable wavelet filter selected. The number of arithmetic operations is reduced with lifting scheme approach. In distributive arithmetic algorithm, the multipliers are replaced with memory operations, pre-computed partial products are computed based on distributive logic and are stored in intermediate memory. The input data is used as address to access the pre-stored partial products and are accumulated to compute the low pass and high pass filter outputs. Both lifting scheme (LS) and distributive arithmetic (DA) scheme can reduce complexity in DTCWT computation, but increases latency and memory space respectively.

[007] In the view of foregoing, there is a need for a method to compute DTCWT that can perform six levels decomposition and also reduce

computation complexity utilizing optimum resources on CLBs and improve processing time for DTCWT computation.

[008] The above mentioned shortcomings, disadvantages and problems are addressed herein, as detailed below.

C) OBJECT OF THE INVENTION

[009] The primary object of the embodiments in the present invention is to provide a method for DTCWT computation using modified DA algorithm with optimum use of CLB resources on FPGA for image fusion applications with reduced time delay in latency and processing time.

[0010] These and other objects and advantages of the embodiments herein will become readily apparent from the following detailed description taken in conjunction with the accompanying drawings.

D) SUMMARY OF THE INVENTION

[0011] The aspects of this invention are directed towards programmable structure for computation of DTCWT. In one embodiment, DTCWT operation consists of decomposition of input image signal into six sub bands at each level and requires J level decomposition (J is six for image fusion). At each level of decomposition the architecture consists of number of processing elements and these processing elements decreases as the computation moves from level 1 to level J. Distributive arithmetic algorithm is one of the processing modules that reduces computation complexity for DTCWT computation.

[0012] The four stage DTCWT computation unit shown in Figure 2 consisting of H-tree and G-tree is realized using a four stage pipelined structure in order to improve the throughput. The novel architecture designed is used for computation of both forward DTCWT and inverse DTCWT and hence is reconfigurable. Every stage of filtering consists of an input buffer that stores one sample of input x(n), the SISO register is sequentially loaded and the three stage arithmetic computation unit processes the input data to generate the low pass output sample.

[0013] The filter bank structure designed consists of a de-multiplexer at the output stage. The first output of de-multiplexer when enabled the filter bank structure can be used for inverse DTCWT computation (shown in Figure 3). The output of filter bank structure is averaged with high pass filter output and the averaged data is taken to next processing stage. If the second output of de-mux is enabled, the filter bank structure is sued for forward DTCWT computation.

[0014] The output of filter bank structure is processed by next phase of low pass and high pass filters. The total time duration T_{4final} for one stage DTCWT computation is increased by T_{avg} delay with the inclusion of de-mux and output adder. The control logic is designed to achieve data synchronization between sub modules and synchronization with four stage data processing. The control unit is also used to reconfigure the designed model for DTCWT or inverse DTCWT computation.

[0015] Figure 4 shows the novel architecture for DTCWT and inverse DTCWT computation realized with reconfigurable filter bank structure.

[0016] The control unit is designed to provide data synchronization between multiple stages of DTCWT computation. At the end of T_{4final} the de-mux is enabled or disabled based on the reconfiguration logic. At the transmitter to perform OFDM de-mux is enabled to direct the input to the output stage operation for averaging with high pass filter output. During OFDM demodulation demux output is reconfigured to direct the output to second stage LPF and HPF filters. Every stage of DTCWT computation is designed with intermediate register so as to achieve pipeline architecture. Each pipeline stage is designed to have a delay of less than $T_{4final}/4$.

[0017] The latency of the pipelined stage DTCWT computation is T_{4final}, and the throughput is 1, i.e after every T_{4final} delay the first output sample is generated and the delay between successive output samples is $T_{4final}/4$. For inverse DTCWT, the latency is $T_{4final} + T_{avg}$ and through put is 1. The control unit ensures the pipeline operation of every stage and also parallel processing operation of all four stages. The latency in the final output computation at the end of four stage is $4*T_{4final}$ and throughput is 1 (delay between successive output samples is $T_{4final}/4$).

[0018] In order to evaluate the performance of the proposed DTCWT and inverse DTCWT architecture, an example set of data with date size

length of 32 is considered. The data is considered to have four frames of symbols with symbol size 8. Each of these frames are serially loaded into the DTCWT computation unit. The outputs of DTCWT obtained are loaded in parallel into the inverse DTCWT structure. The control unit at the DTCWT and inverse DTCWT structures are designed to synchronize the data movement operation. The output of inverse DTCWT are recorded to compare with the input data and logic correctness in verified. Test vectors are either +1/-1, at the output of inverse DTCWT the output is 9 bit integer, a threshold value is set to convert the 9 bit number to two bit data representing +1/-1.

[0019] In this invention, the four stage DTCWT architecture shown in Figure 4 consists of LUT model that is realized using modified DA algorithm.

[0020] In this invention, general structures based on DA and multiplexers are introduced for computation of DTCWT. The advantages of DA algorithm are combined with the advantages of multiplexer algorithm to ensure optimum use of FPGA resources for computation of DTCWT.

[0021] According to one embodiment of the present invention, the architectures designed are independent of image size, number of levels and are customized to filters of length 10. The architecture can be implemented with pipelining and parallel processing modules to further improve processing time with trade-off between area, speed, cost and power dissipation. The invention provides architectures that are regular and easily extendable for higher order

filters and does not contain feedback and long paths that introduce latency. The architecture is best suitable for FPGAs and also can be implemented using systolic structures.

[0022] According to one embodiment of the present invention, the method has a reduced time delay of 2 ns for level one computation of DTCWT coefficients.

[0023] According to one embodiment of the present invention, the method implements a hardware architecture comprising of DA elements with improved throughput of 50% for DTCWT computation.

[0024] According to one embodiment of the present invention, the hardware architecture has maximum operating frequency of 497 MHz with power dissipation reduced to less than 2 W.

[0025] These and other aspects of the embodiments herein will be better appreciated and understood when considered in conjunction with the following description and the accompanying drawings. It should be understood, however, that the following descriptions, while indicating preferred embodiments and numerous specific details thereof, are given by way of illustration and not of limitation. Many changes and modifications may be made within the scope of the embodiments herein without departing from the spirit thereof, and the embodiments herein include all such modifications.

E) BRIEF DESCRIPTION OF THE DRAWINGS

[0026] The other objects, features and advantages will occur to those skilled in the art from the following description of the preferred embodiment and the accompanying drawings in which:

[0027] Figure 1 illustrates the filter bank structure of DTCWT according to one embodiment of the present invention

[0028] Figure 2 illustrates DTCWT filter bank structure, according to one embodiment of the present invention

[0029] Figure 3 illustrates the modified architecture for DTCWT computation, according to one embodiment of the present invention

[0030] Figure 4 illustrates the four stage DTCWT computation, according to one embodiment of the present invention

[0031] Figure 5 illustrates the multiplexer based DTCWT filter computation based on symmetric logic of filter coefficients, according to one embodiment of the present invention

[0032] Figure 6 illustrates the hybrid DTCWT filter architecture combining multiplexer based logic and DA logic, according to one embodiment of the present invention

[0033] Figure 7 illustrates the hybrid architecture for high pass filter bank for DTCWT computation, according to one embodiment of the present invention

[0034] Figure 8 illustrates the DTCWT architecture for first stage 2 D DTCWT computation for image of size 100 x 100 with data rearrangement, according to one embodiment of the present invention

[0035] Figure 9 illustrates the top level architecture of 2D DTCWT computation, according to one embodiment of the present invention

[0036] Figure 10 illustrates the 2D IDTCWT computation synthesis netlist, according to one embodiment of the present invention

[0037] Table 1 illustrates the low pass and high pass filter coefficients that are scaled and represented in 2's complement format, according to one embodiment of this invention

[0038] Table 2 illustrates the comparison results of three different schemes for DTCWT computation, according to one embodiment of the present invention

F) DETAILED DESCRIPTION OF THE PREFERREED EMBODIMENTS

[0039] The 2D DTCWT architecture (shown in Figure 1), that consists of four filters represented by La, Lb, Ha and Hb in the first stage. The first stage processes the input signal, by computing DTCWT on the rows. The second stage consists of eight filters that process along the columns to compute eight sub bands. From eight sub bands two are low pass sub bands (LLR1, LLC1) and the rest six of them are high pass sub bands (LHR1, LHC1, HLR1, HLC1, HHR1 and HHC1).

[0040] The filter coefficients of DTCWT (Shown in Table 1) for stage 1 low pass and high pass consist of 10 filters, each of these filters are scaled by 256 and round to its nearest integer and is represented in 2's complement format.

[0041] The filter coefficients are predefined and depend upon corresponding wavelet selected. The selected wavelet filter for DTCWT consists of 18 low pass and 10 high pass filter coefficients. If the image size is N x M (N is the number of row pixels, M is the number of column pixels), it is required to perform N 1D DTCWT row wise and M 1D DTCWT column wise. Thus it is required to perform N+M 1D DTCWT computation for each filter bank.

[0042] Each 1D DTCWT computation on the N rows requires low pass filtering and high pass filtering. Computation of every output of low pass filter requires 18 multiplications and 17 addition operations as there are 18 filter coefficients for low pass filtering. As there are M pixels in each row, 1D DWT computation of each row requires 18M multiplications and 17M additions. As there are N rows thus it is required to perform 10NM multiplications and 9NM addition operations. Similarly for computing high pass filter coefficients for the row elements it requires 18NM multiplications and 17NM addition operations.

[0043] Computation of 1D DTCWT on the columns also requires 18NM multiplications and 17NM additions for low pass and 18NM multiplications and 17NM additions for high pass. Thus for N x M image the total number of multiplications are 2(18NM+18NM) and 2(17NM+17NM) additions for level 1 decomposition for real tree alone. N x M size image after low pass and high pass filtering and down sampling the data size reduces to N/2 x M/2 and there are four sub bands.

[0044] Level 2 decomposition operates on LL sub band which is of size N/2 x M/2. The number of multiplications and addition operations for level 2 is (18NM+17NM) and (18NM+17NM) respectively. Thus for computation of i^{th} level DWT the number of multiplications and additions required are $2(18NM+17NM)*2^{-j}$ and $2(18NM+17NM) *2^{-j}$, where j = 0, 1, 2 3, ….i, for low pass and high pass respectively for real tree alone.

[0045] The total number of multiplications and additions for computation of i-level real DTCWT of input image of size N x M using 18/18 wavelet filter is as given by,

$$(\sum_{j=0}^{i} 2(18NM + 18NM)*2^{-j})_{\text{multiplications}} + (\sum_{j=0}^{i} 2(17NM + 17NM)*2^{-j})_{\text{additions}}$$

$$\text{for } j = 0,1, 2, \ldots i.$$

[0046] In addition to real tree in level one, there is imaginary tree also, which also requires equal number of multiplication and additions operations. In 2D DTCWT computation, there are is another imaginary tree, thus the number of arithmetic operations for DTCWT computation is twice that of DWT. In order to reduce the computation complexity DA algorithm is proposed and is modified to optimize area and timing with maximum utilization of FPGA resources.

[0047] The modified DA algorithm is with combined terms that are in power of 2, the above equation can be written as in (8),

$$Y_{La} = -\left[b_{10}L_{a1} + b_{20}L_{a2} + \ldots\ldots b_{100}L_{a10}\right]$$

$$+\left[\begin{pmatrix}(b_{11}L_{a1})\\+\\(b_{21}L_{a2})\\+\\.\\.\\(b_{101}L_{a10})\end{pmatrix}2^{-1} + \begin{pmatrix}(b_{12}L_{a1})\\+\\(b_{22}L_{a2})\\+\\.\\.\\(b_{102}L_{a10})\end{pmatrix}2^{-2} + \ldots \begin{pmatrix}(b_{1(N-1)}L_{a1})\\+\\(b_{2(N-1)}L_{a2})\\+\\.\\.\\(b_{10(N-1)}L_{a10})\end{pmatrix}2^{-(N-1)}\right] \ldots\ldots(8)$$

[0048] The above equation can be compressed and written as follows, and can be mathematically simplified to equation (9),

$$Y_{La} = \sum_{K=1}^{10} b_k L_{ak} + \left[\left(\sum_{k=1}^{10} b_{k1} L_{ak} \right) 2^{-1} + \left(\sum_{k=1}^{10} b_{k2} L_{ak} \right) 2^{-2} + ... \left(\sum_{k=1}^{10} b_{k(N-1)} L_{ak} \right) 2^{-(N-1)} \right]$$

$$Y_{La} = -\sum_{K=1}^{10} b_k L_{ak} + \sum_{n=1}^{N-1} \left[\sum_{k=1}^{10} b_{kn} L_{ak} \right] 2^{-n} .....(9)$$

[0049] Simplified revision of equation (8) is represented in equation (9) and it is termed as DA algorithm. From the equation number (8) all the terms of the coefficients L_{a1} , L_{a2} ..... L_{a18} are being multiplied by corresponding bits from input X. The term b11, b21, ... b101 represent MSB of input X, $b_{1\,(N-1)}$, $b_{2\,(N-1)}$ .... $b_{10(N-1)}$ represented LSB of input X.

[0050] The input X can be of any number with $2^{(N-1)}$ possible combinations. As per equation (8) the LSB of all the ten inputs of X are multiplied by La1..... La18 coefficients and are accumulated. As there are 10 bits of $b_{1\,(N-1)}$, $b_{2\,(N-1)}$ .... $b_{10(N-1)}$ the multiplication will lead to the 2^{10} possible partial product of h_{a1}, h_{a2}..... h_{a10}, for example if $b_{1\,(N-1)}$ ..... $b_{10(N-1)}$ are [0,0,0,0,.....1] then the multiplication will lead to a partial product of L_{a10}. Thus a look up table (LUT) can be designed to store pre computed partial products and the inputs X is used as address to access the partial products from the memory.

[0051] The limitations of DA algorithm are that it has a latency of 188 clock cycles and throughput of 36 clock cycles. The latency cannot be avoided as it requires initial loading of 10 input samples each of 8-bits content into the register array. However, throughput can be reduced from 36 clock cycles to 8 clock cycles. The modified DA architecture consists of input array register that is serially loaded, however, the address for LUT are the MSBs of input register instead of LSBs.

[0052] Further the MSBs that are red out of input register are simultaneously loaded into the next stage input register from the LSB. Also the new set of input sample is also serially loaded into the input register array from the LSB. Thus at the end of 88 clock cycle, output is computed, and at the same time new input is also loaded into the input register array. Thus the latency is reduced from 16 clock cycles to 8 clock cycles. The right shift register is replaced with left shift register.

[0053] The DTCWT filter bank can also be implemented using the multiplexer logic (shown in Figure 3). The multiplexer architecture consists of input register array (10 registers as there are 10 filter coefficients), multiplexer array and the adder/accumulator array.

[0054] From the symmetric property of filter coefficients, it is required to reduce the number of multiplications from 10 to 4. Prior to multiplication of inputs with the filter coefficients, the inputs are added/subtracted with the adder/subtractor array. Thus the number of multipliers is reduced from 10 to 4 (shown in Figure 3).

[0055] In the multiplexer based logic, the LSBs of each register is used to select the filter coefficient which is either 0 or the filter coefficient itself. The outputs of multiplexers are accumulated and are finally added in the adder array.

[0056] The primary building blocks of Xilinx FPGA are configurable logic blocks (CLB), programmable interconnecting array and I/O devices. The Xilinx Spartan 3 FPGA CLB is made up of two slices and each slice consists of two logic elements. Each logic element consists of one four input LUT, carry & control logic and flip flop. The four input LUT can be used to realize 64K combinational

or sequential functions, or can be used as 64K one bit memory, or can be used as 16 bit shift register.

[0057] The carry and control logic is made up of multiplexers and is used as fast carry chain. The synthesis tool optimizes the HDL logic to LUT based logic and implements the logic using LUT resources. Dedicated multipliers and memory resources are also available on the FPGA that can be used for specific purpose such as DSP applications. Predominantly, Xilinx synthesis tools use LUT resources for logic implementation. The multiplexers that are available within the CLBs can also be used depending upon user requirements; by default the synthesis tool maps the HDL logic onto LUTs.

[0058] In order to map the DTCWT architecture on CLBs, it is required to use both LUTs as well as multiplexers. The present invention addresses this by splitting the DA equation horizontally into two half. The upper half of equation (8), is realized using multiplexer logic and the lower half of DA algorithm is realized using modified DA algorithm.

[0059] The scaled filter coefficients have a maximum integer value of +/- 211 and hence are represented by 9-bit 2'c complement integer format number. The convolution operations of input symbol with filter coefficients are represented as in Equation (15) and Equation (16) for low pass and high pass respectively. Filter coefficients are represented by H_{oa} and H_{1a}.

$$L(n)=\sum_{k=0}^{N-1}H_{0a}(K)\,X(n-K),\qquad n=0,1,2,.....,N-1 \quad \ldots (15)$$

$$H(n)=\sum_{k=0}^{N-1}H_{1a}(K)\,X(n-K),\qquad n=0,1,2,.....,N-1 \quad \ldots (16)$$

[0060] For n=18, the convolution operation in Equation (15), can be expressed as in Eq. (17),

$$L(17)=\sum_{K=0}^{17} H_{0a}(K)\,X(17-K) =$$

$$\mathrm{L}(17) = H_0(0)X(17) + H_0(1)X(16) + \cdots + H_0(17)X(0) \cdots (17)$$

[0061] From Table 2, for bior6.8 wavelet filter, filter coefficients $H_{0a}(1)$, $H_{1a}(0)$, $H_{1a}(1)$, $H_{1a}(2)$, $H_{1a}(14)$, $H_{1a}(15)$, $H_{1a}(16)$, and $H_{1a}(17)$ are zeros, and there exists symmetric between remaining low pass filter coefficients. Equation (17) can be reduced to Equation (18) by extending the symmetric property

$$\mathrm{L}(17) = H_0(1)[X(16) + X(0)] + H_0(2)[X(15) + X(1)] + \cdots + H_0(9)X(8) \cdots (15)$$

[0062] The computation of sample L(17), requires 9 multipliers instead of 18 multipliers, and the number of adders required is 16, out of which 8 adders are required to add the QAM symbols X(n) prior to multiplication of filter coefficients, another 8 adders add the multiplied partial products. Similarly, computation of every sample in Equation (16) requires 9 multipliers and 16 adders. With the incorporation of symmetric property, for computation every sample of L(n) and H(n), the number of multipliers are reduced from 18 to 9, and number of adders required are 16. The computation complexity of DTCWT with symmetric structure is expressed as $CCB = 4O\{(\frac{\alpha}{2}) + (\beta - 4)\}$), for four stage computation of DTCWT the computation complexity is reduced by 28%.

[0063] The filter structure of DTCWT with symmetric logic is designed, the filter structure designed consists of four stages, the first stage is SISO register, second stage is array of adders for adding input data, the third stage is the multiplier and the last stage is the adder array for final accumulation of partial products. The input data that are converted to parallel data and stored in the intermediate register enter the SISO register (internal register of DTCWT filter structure). The SISO output data are loaded into the adder

array for addition operation as per the logic in Equation (17), the output of adders are multiplied with the corresponding filter coefficient, the partial products from the multiplier are accumulated at the adder array as shown in Figure 3 to generate the DTCWT coefficients. The SISO consisting of 17 registers, feed data into the adder array consisting of 8 adders that accumulate DTCWT coefficients.

[0064] As there are four filters in every stage, a pipelined architecture is designed as shown in Figure 4. The pipelined architecture consists of LUT that is designed using DA algorithm.

[0065] Prior to splitting the DA equation into two half, symmetry property of wavelet coefficients are incorporated to reduce memory space. The modified DA (termed as Hybrid DA) is a combination of DA and multiplexer logic

[0066] The eight bit input data is serially loaded into the input register, loading of data into 18 registers array requires 180 clock cycles. An enable signal is used to acknowledge the loading of input register, at the 181^{st} clock cycle, addition and subtraction operations are carried out and the results are stored into four intermediate registers that are of nine bit in length. Addition of two eight bit numbers increases the accumulated output to nine bits.

[0067] The four intermediate registers are designed as shift registers, and are interconnected to form a 136-bit shift register. The four registers are loaded parallelly from the adder/subtractor at the end of 81st clock cycle. To compute the partial products, the top two registers are connected to multiplexers and the bottom two registers are connected to LUTs. The MSB bits of top two registers are used as select line to the top two multiplexers. Similarly the bottom two

registers MSBs are used as address to LUT memory. The LUT consists of 4 memory locations (the address is two) and each space hold eleven bit partial products.

[0068] Thus the present invention reduces the memory space from 1024 memory locations to 4 memory locations. The hybrid architecture utilizes both the LUT space as well as carry & control logic space existing on logic elements within the CLB.

[0069] Hybrid logic combining multiplexers with DA logic is also used in designing all the four sub band filters with filter coefficients La, Lb, ha and Hb. The hybrid architecture for high pass filter is shown in Figure 7. The registers array is organised as per the filter coefficients and the hybrid logic requires an LUT and two multiplexers.

G) ADVANTAGES OF THE INVENTION

[0070] The present invention provides hybrid architecture for DTCWT computation in much faster time with area efficient and higher throughput with respect to the conventional models

[0071] It is to be understood that the phraseology or terminology employed herein is for the purpose of description and not of limitation. Therefore, while the embodiments herein have been described in terms of preferred embodiments, those skilled in the art will recognize that the embodiments herein can be practiced with modification within the spirit and scope of the claims.

CLAIMS:

What is Claimed:

1. A dedicated and customized high speed architecture performing computation of dual tree complex wavelet transform operation, that decomposes input signal such as image (2D Data) into multiple sub bands based on specific number of decomposition levels (supports till six levels) with each level comprising of low pass and high pass filter banks that compute real and imaginary components of the input image, with each filter bank realized using processing elements comprising of basic logic blocks, the basic logic processing blocks operate on input data that are obtained from the preceding stages and the outputs of each stage stored in intermediate memory
2. A dedicated high speed architecture according to claim **1**, where in the input signal vector comprises of N x N input samples and the number of basic processing elements implemented in each of said processing stages of the customized DTCWT architecture decreases every stage as the input is scaled by half as the processing progresses from level 1 to level n.
3. The dedicated architecture according to claim **2** processes all the input signal of N x N in parallel with use of separable property supported by DTCWT
4. The dedicated architecture according to claim **2**, that processes N x N input data with row processor and column processor with use of 2D

DTCWT, with row and column processor comprising of 1D DTCWT processor architecture, each row and column processor architecture comprising of two filter banks of real and imaginary filters, each real and imaginary filter bank comprising of low pass and high pass filter architectures, each low pass and high pass filter

5. The dedicated architecture for DTCWT computation according to claim **4**, computes n-level DTCWT sub bands as per the requirements of image registration.
6. The dedicated architecture for DTCWT computation according to claim **4**, comprises of real and imaginary filter banks with each filter bank consists of low pass and high pass filters with each filter architecture supporting 18 filter coefficients that are scaled by 256 and rounded to nearest even number
7. The dedicated architecture according to claim **6**, is designed to operate with even numbered filter coefficients as the number of bits that are '1' is minimum as they are represented with 2's complement number system and LSB is always '0', as the coefficients are rounded to even number thus reducing power dissipation
8. The dedicated architecture for DTCWT computation according to claim **3** comprises of an input memory, intermediate memory at the output of 1D DTCWT to store the DTCWT row coefficients, another intermediate memory at the output of 2D DTCWT processor to store the sub bands. The input memory, two intermediate memory and output

memory are controlled by FSM based control logic for data read, data rearrangement and data storage. The FSM control unit is designed to read the data row wise from input memory into the 1D DTCWT processor architecture consisting of four filters, the output of the sub bands are rearranged and stored into four intermediate output memories. The second stage FSM control unit controls data movement from four intermediate memories into DTCWT architecture consisting of eight sub bands processors into eight output memories.

9. The dedicated architecture for DTCWT computation according to claim **8,** has an output stage consisting of three butterfly data flow graph with adders and subtractors and scaling operation with 1/sqr(2). The scaling function whose numerical value is 0.707106 is approximated to 0.75(which is equal to 0.5 + 0.25), thus the data at te output is right shifted by once and also twice, both of these data re accumulated to obtain the final scaled output. The proposed logic eliminates use of multipliers for scaling operation and thus improves computation time and reduces logic cells.
10. The 2D DTCWT dedicated architecture consists of 12 filter banks, four in first stage, eight in second stage. Each of these processing units are realized using hybrid architecture comprising of multiplexer logic and DA logic. The symmetric property of filter coefficients in all the 12 filters is utilized thus reducing the number of input registers from 10 to 4, improving 60% saving in register memory.

11. The method as claimed in claim 10, due to symmetry property there are two stages of registers arrays that are separated by arithmetic operation such as adder/subtractor. These two stages of register array are designed with pipelined logic and are operated sequentially. The first stage is 8-bit register and 18 register arrays, second stage is 9-bit register with four register arrays. A control unit that control pipeline operations is designed to load and shift data into the two register arrays with 1T clock delay, thus reducing delay by 1 clock cycle.
12. According to the embodiment in Claim 11, 2D DTCWT that comprises of 4 filter in the row processing stage and 8 filters in the column processing stage and in total comprising of 12 filters for level-1 2D DTCWT computation is designed using only 8 filters. With four filters for row processing and another four filters for column processing.
13. According to the embodiment in Claim 12, with each filter comprising of 18 filter coefficients with symmetry property the filter structure is a folded structure design and requires only 9 filter coefficients.
14. According to the embodiment in Claim 13, each of the four filters required for row processing and column processing is realized using a four stage parallel structure that processes data in parallel to generate the required DTCWT coefficients.
15. According to the embodiment in Claim 11, a four stage pipelined structure is designed to generate four level DTCWT coefficients. With

each of the four stage pipelined structure comprising of DA logic for FIR filter design.

16. The method as claimed in claim 15 comprising of 12 filter banks, each of them realized using hybrid logic that consists of multiplexers based multiplication and DA algorithm.
17. The method as claimed in claim 16, the DA based logic is designed such that two register arrays MSBs are pointed to LUT memory instead of LSB, the LUT contents are read and are accumulated at the output and the accumulated contents are right shifted (instead of performing left shift operation). At each clock cycle, the MSBs that are read out forming the address of LUT are also stored into the LSBs of preceding register array thus improving the throughput by 50%.
18. The method as claimed in claim 15, consists of four stages of pipeline structure: first stage array, second stage array, multiplexer-DA logic with accumulation and final adder stages. FSM control unit is designed to control data flow into each of these stages so as to achieve throughout of 8 clock cycles, which is an improvement of 50% than the classical techniques.
19. The dedicated architectures as claimed in claim 15, consisting of 12 filter banks, each of them with different set of filter coefficients are designed with the hybrid logic as claimed in claim 11, are customized as per the filter coefficients. A master control unit synchronizes the data movement among all the filter banks.

20. The 2D DTCWT architecture designed is reconfigurable with regard to input data size. The dedicated architecture is designed to operate on input data of N x N, six stages of decomposition is required for image fusion, the dedicated architecture is designed to operate on data size ranging from N x N to N/32 x N/32.
21. The 2D DTCWT architecture is designed to perform inverse DTCWT with reversal of filter coefficients in each of the 12 filter banks. The 12 filter banks designed for computation of forward DTCWT are rearranged to compute inverse DTCWT, thus the filter bank structures are reusable for computation of both forward and inverse DTCWT.
22. The hybrid architecture as claimed in claim 11, operates at maximum frequency of 497 MHz with power consumption less than 0.21 W, and LUTs less than 128 numbers.
23. The level-1 2DDTCWT as in Claim 12, operates at maximum frequency of 321 MHz consuming less than 1 W of power and occupies less than 1200 LUTs on Virtex-5 FPGA.

ABSTRACT

The present inventions disclose n-level of decomposition using DTCWT, with each level comprising of 4 filter banks, each filter bank realized suing hybrid logic. Four stage parallel structure integrated with four stage pipelined structure design computes 2D DTCWT coefficients that are required for image registration process. Each of the filter structure is designed using hybrid logic. Hybrid logic is design with combination of multiplexer based multiplier and distributive arithmetic algorithm. The distributive arithmetic algorithm is designed to improve throughput by 40%. The method comprises of pipelined stages for computing DTCWT, the DTCWT filters banks are reusable for computing inverse DTCWT. The 2D DTCWT is scalable and can operate on images of size ranging from N x N to N/32 x N/32. The method comprises of capturing two digital images from two different sources, six levels of decomposition is computed and from the sub bands appropriate ones are selected for image registration. The operating frequency of 2DDTCWT is 321 MHz and is very much suitable for real time applications.