CHAPTER- 2

LITERATURE REVIEW

Down-sampled low-pass and high-pass filter computation is performed repeatedly on the data sequence to generate sub-band components of multi-level DWT. Due to variation in sampling rate, mapping DWT algorithm into a hardware design is not straight-forward. Algorithm modification and architectural design are considered to make the algorithm mapping simple and find an efficient hardware design. During last two decades, different computation schemes and several architectural designs schemes have been proposed in the literature to develop an efficient hardware design for 2-D DWT. These designs are broadly divided into four types. These are (i) PA and RPA based designs, (ii) folded-based designs, (iii) parallel designs and (iv) multiplier-less designs. A literature review is made to find the research progress and identify the design issues to develop an efficient hardware design for 2-D DWT.

2.1 PA AND RPA-BASED DESIGNS

Conventionally, DWT computation is performed using convolution approach. Both orthogonal as well as bi-orthogonal wavelet filter computation are performed using linear convolution. Lewis and Knowles (1991) have proposed VLSI architecture for DWT based on convolution scheme. They have used multiplier-less design method to derive the architecture. Parhi and Nishitani (1993) have proposed two architectures that combine the world-parallel and digit-serial methodologies. Separate filtering units are used for parallel computation of *J*-level DWT computation. Due to down-sampled filter computation, the hardware utilization of the filtering unit decreases progressively by factor 2. Vishwanath (1994) proposed the RPA based design to improve HUE in DWT design. Subsequently, few RPA-based structure have been proposed for efficient implementation of multi-level DWT. Grzeszczak *et al.* (1996) have proposed a systolic architecture for multi-level 2-D DWT using one filtering unit and complete *N*-point DWT in 2*N* computational cycles. Chakrabarti and Mumford (1996) have proposed another RPA-based 2-D DWT structure consisting of 4 parallel filters. Limqueco and Bayoumi (1998) have proposed a RPA based design using six FUs. Four FUs are used to perform the first level DWT computation, while the remaining two FUs are used to perform the higher level DWT computation using RPA.

Denk and Parhi (1998) have proposed a systolic structure that requires two filtering units and perform *N*-point DWT in *N* computational cycles. Marino *et al* (2000) have proposed hybrid cascaded structure comprising of one PA unit and one RPA unit for implementation of multi-level 2-D DWT. The PA unit performs computation of first level DWT while the RPA unit performs DWT computation of all higher levels. Consequently, the HUE of the RPA based 2-D DWT structure is 87.5% which is higher than the previously proposed similar designs. In general, these designs are focused on systolization, throughput rate and hardware utilization to develop an efficient hardware design for 1-D and 2-D DWT.

The lifting scheme is proposed by Swelden (1996), which is also known as second-generation wavelet. The lifting scheme has several useful properties like symmetric forward and inverse transform, in-place computation and integer-to-integer wavelet transform. The lifting-based DWT structure required much less number of multipliers, adders and storage elements compared to the convolution-based algorithm. The above features make the lifting scheme more appropriate for DWT implementation. Several architectures have been suggested for implementation of 1-D and 2-D DWT to take advantage of lifting scheme. Huang *et al.* (2001) have suggested architecture for computation of 1-level lifting DWT. This structure requires 2*N* clock cycles to complete an *N* point lifting DWT. Subsequently, RPA-based architecture has been proposed by [Chen (2004)] for Multi-level lifting DWT. Few designs also have been proposed for 2-D DWT using lifting scheme. Liao *et al.* (2004) have proposed dual scan architecture for lifting 2-D DWT where two samples are processed in every clock cycle. Tseng *et al*. (2002) and Xiong *et al*. (2006a) have proposed separate hybrid architecture for lifting 2-D DWT where RPA computation is performed in separate pipeline stages. The first stage of the hybrid design computes the first level 2-D DWT while the second stage computes all the higher DWT levels using RPA. The HUE of this structure is nearly 89%, which is higher than the previous proposed structures.

2.2 FOLDED DESIGNS

A PA-based 2-D DWT design uses simple control circuitry but offers significantly less utilization efficiency. On the other hand the RPA-based design offers higher utilization efficiency than the PA-design but involves complex control circuits but the HUE of RPA design

is far less than the maximum value. To overcome these difficulties, line-based folded architecture is suggested by [Wu and Chen (2001)] for convolution based 2-D DWT, where the poly phase decomposition of filter coefficient and folding of multi-level DWT computation are used to increase the throughput rate of the structure. The 2-D DWT structure based on folded scheme requires small on-chip memory than the RPA and PA designs, simple control circuits and offers 100% utilization efficiency. However, the folded design requires frame-buffer to perform multi-level 2-D DWT computation in level-by-level. Due to simplicity of design and higher utilization efficiency, folding scheme extensively used during the last decade to develop efficient hardware designs for Multi-level 2-D DWT. Andra *et al.* (2002) have proposed single processor architecture for lifting 2-D DWT. This structure is more generalized and can compute both the 2-D forward and inverse transform using different bi-orthogonal filters. Dai *et al*. (2004) have also suggest a line-based folded 2-D DWT structure to increase the throughput by twice by nearly doubling the multiplier, adder and increasing the on-chip memory size by 3*N* words. Subsequently, Huang *et al*. (2005) have also proposed a line-based folded structure for convolution-based 2-D DWT which requires less on-chip memory than the structure of [Wu and Chen (2001)]. Liao *et al*. (2004) have proposed one lifting-based folded architecture for lifting 2-D DWT. Few lifting-based 2-D DWT designs also have been proposed by [Barua *et al*. (2005), Lan *et al*. (2005) and Seo and Kim (2007)] where the critical path of lifting structure is reduced by pipelining which increases the overhead complexity. Few flipping based designs also have been proposed for efficient implementation of 2-D DWT with less critical-path delay [Huang *et al*. (2002), Huang *et al.* (2004) and Xiong *et al.* (2006), Lai *et al.* (2009); Zhang *et al.* (2012)]. All these structures differ by arithmetic complexity, pipeline registers, critical-path delay and throughput rate. Few block-based structures also have been proposed for high-throughput and area-delay efficient realization of lifting 2-D DWT [Tian *et.al* (2011), Mohanty *et al.* (2012)]. Maye and Srinivasulu (2015) have proposed lifting based folded 2-D DWT structure. Wang and choy (2016) proposed a systolic array based 2-D DWT architecture using overlapped data scanning method to reduce the on-chip memory.

2.3 PARALLEL DESIGNS

The folded architecture uses one filtering-unit (FU) and calculates multi-level 2-D DWT in level-by-level serially using a frame-buffer. The most advantage part of folded architecture is the simplicity of the design, small arithmetic complexity and offers 100% hardware utilization efficiency (HUE). But, the frame-buffer which is external to the chip and that affects significantly the performance of folded architecture. On the other hand, the RPA-based architecture requires one or two filtering units and performs Multi-level DWT computations concurrently without using frame-buffer. However, the RPA-based design involves nearly two times more on-chip memory words than the folded design and involves complex control circuitry. Besides, the RPA-based design offers HUE less than 100% HUE. The parallel architecture is a compromise between the folded architecture and the RPA-based architecture. The parallel architecture involves separate filtering units for each decomposition levels with reduced complexity. It does not required frame-buffer unlike the folded architecture but involves higher on-chip memory and arithmetic resources. However, the parallel architecture requires less on-chip memory words than the RPA-based structure and offers 100% HUE. Besides, the parallel architecture offers throughput-scalability. An additional filtering unit when integrated at the front-end of the parallel architecture then the design is configured for next higher DWT level and the throughput rate is enhanced by 4 times [Mohanty and Meher (2011)]. To take advantage of these features, in recent years a few parallel architectures have been proposed for implementation of multi-level 2-D DWT using less on-chip memory. A lifting-based parallel structure is proposed in [Mohanty and Meher (2011)]. This structure uses interfacing units to feed the intermediate data-blocks to the succeeding filtering units (FUs) in folded form in order to achieve 100% HUE in the design. Mohanty and Meher (2013) have proposed parallel structure based on convolution scheme. This structure does not require interfacing units unlike the structure of [Mohanty and Meher (2011)] and requires less on-chip memory. However, the structure of [Mohanty and Meher (2013)] involves more arithmetic resources than the [Mohanty and Meher (2011)] for same throughput implementation and introduces some overhead complexity to the input-buffer. Subsequently, Hu and Jong (2013) have proposed a scalable parallel architecture for multi-level 2-D DWT based on flipping scheme. This structure requires less on-chip memory and critical path delay compared to earlier structure. However this structure

also required interfacing circuit for data multiplexing 2-D DWT computation. Hu and Jong (2015) have proposed memory-efficient parallel 2-D DWT architecture. In this structure memory is reduced by word length optimization of internal signal. Zhang *et.al*. (2016) also have proposed parallel architecture for multi-level lifting 2-D DWT using dual scanning scheme where two samples are processed in every clock cycle.

2.4 MULTIPLIER LESS DESIGNS

A regular and modular bit-parallel (word-level) computing structure for DWT easily obtained using multiplier. However, multiplier when implemented in dedicated hardware involves significantly more combinational logic than the adder. A parallel structure of DWT involves large number of multipliers. Consequently, multiplier consumes a significant part of the chip area and power. The number multipliers can be accommodated in the parallel design and the throughput rate depends on the availability of combinational resource in the chip. Silicon-area, speed, power consumption is considered the general performance evaluation parameters while designing a VLSI architecture. Earlier, power consumption was the designer's secondary concern comparison to area and speed. However, in recent years, power is being given comparable or higher weightage to area and speed due to phenomenal growth of portable and wireless handheld multimedia devices. The average power consumption is the most critical design concern for these devices [Parhi (1999)]. As high-performance wireless and portable devices are proliferating, battery technology is reaching its limit, making low-power designs is become more attractive. There are various factors that contribute to the power budget of a VLSI circuit. Combinational logic complexity is one of them. Therefore, low-complexity design most often leads realization of a low-power circuit. Several schemes and methodologies have been proposed and found in the literature for low-complexity design.

Bit-Serial structure processes one-bit input at a time whereas the multiplier-based structures process all the bits of input sample in one clock cycle. The bit-serial structure involves significantly less combinational logic than the multiplier-based structure and processes input samples at nearly *w*-times slower rate than the multiplier-based structure, where *w* is the bit-width of input sample. Few bit-serial structures have been proposed in the literature for low complexity realization of DWT [Martina and Masera (2007)**,** Longa *et al*. (2008), Mohanty and

Meher (2009)]. The DWT algorithm uses constant multiplications. Constant multipliers can be implemented using shift-add method instead of direct use of hard multipliers to take advantage of the fixed bit-pattern of the constant multiplication operand value. In the shift-add method, the multiplication of a variable $\{x\}$ with the constant value $\{c\}$ is represented as a sum of shifted version of $\{x\}$, where the number of addition operation depends on the number of logic '1' in the 2's complement representation of the constant value. The shifting operation is implemented through hard-wiring. Therefore, the combinational logic complexity of constant multiplier based on shift-add method depends on the adder complexity. Different types of number systems have been considered to represent the constant values using less number of logic '1's. Canonical sign digit (CSD) is popularly used for representation of constant values [Hartley (1996)]. CSD representation uses 33% less logic '1's than those required by 2's complement representation of any signed decimal value. The convolution based DWT computation involves inner-product computation. An N-point inner-product computation involves N constant multiplications. These constant multiplications are performed in a single stage in parallel and the addition operations are performed in a separate stage. In other word, a set of N constant multiplication operations are performed together in the first stage of computation of N-point inner-product computation. The adder complexity of a set of constant multipliers can be reduced by using multiple constant multiplication (MCM) approach. Therefore, CSD representation of constant values and using MCM approach the combinational logic complexity of inner-product design can be reduced substantially than using hard multipliers without compromising on the throughput rate. Few MCM-based low-complexity designs also have been proposed in the literature for implementation of 1-D DWT [Martina and Masera (2007) and Mohanty *et al* (2015)]. Martina and Masera (2007) have approximated 9/7 filter coefficients and expressed 9/7 filter outputs in terms of 5/3 filter output using common multiple constant method. They have also suggested a multiplier less structure to compute both 5/3 and 9/7 filters based DWT. The structure of [Martina and Masera (2007)] involves significantly less adders then the earlier structures.

According to the International Technology Roadmap for Semiconductors (ITRS), the memory capacity is increasing faster and it is higher by several orders of magnitude then the logic capacity in the chip. It can, therefore, be expected that it would be possible to have much more memory elements in a chip than logic elements. Traditionally memory has remained as an

integral part of system to store the secondary data. The concept of memory as a standalone subsystem is no longer valid. Memories are integrated as part within the processor chip to derive higher memory bandwidth between a processing unit and a memory unit with much lower power consumption [Furuyama (2004)]. Memory-based designs are more regular compared to the logic-only based designs. Besides, the memory-based design has many other advantages such as greater potential for high-throughput and reduced-latency implementation, (since the memory-access time is much shorter than the usual multiplication-time) and are expected to have less dynamic power consumption due to less switching activities for memory-read operations compared to the conventional logic-only based designs. Therefore, memory-based computing system would be a promising alternative to the conventional logic-only based implementation. An appropriate combination of logic-based arithmetic circuits and memory-based computing elements may be integrated together for dedicated implementation of DSP functions. Therefore memory-based designs for implementation of arithmetic operations become increasingly popular in recent years [Meher (2007)]. The memory-based designs are broadly divided in two types; (i) direct memory-based and (ii) distributed arithmetic (DA) based. In direct-memory based design, multiplier is implemented using look-up tables (LUT). The LUT stores 2^w possible values of the product of w-bit multiplier operand with a constant multiplicand operand. The w-bit multiplier operand fed to the address line of the LUT to retrieve the multiplication result. In DA, multiplication operations of inner-product computation are lumped and performed together. Therefore, DA helps to develop regular structures for various signal processing algorithms. White (1989) developed a mathematical formulation to perform inner product computation using DA approach. The DA structure consists of a LUT and accumulator. The LUT stores 2^K possible inner-product values of N-point constant vector with N-point bit-vector. The bit-vector is fed to the address line of the LUT to retrieve the inner-product values (of the constant vector with the corresponding bit-vector available at the address line) from the LUT. Bit-slices (bit-vectors) of an K-point input-vector are fed to the DA LUT serially in least significant bit (LSB) to most significant bit (MSB) order to retrieve the partial inner-product values. The partial inner-product values of successive bit-slices of input-vector are shift-added in accumulator to produce complete value of inner-product computation. The computation time (to complete the inner-product computation) depends on the word-length (w) of the input-vector, LUT access time (T_{MR}) and adder delay (T_A). The DA structure also considered to be a bit-serial structure, since

the partial inner-product values are accumulated serially in the accumulator in *w* bit-cycles, where one bit-cycle is defined as $T = T_{MR} + T_A$. Few DA based designs have been proposed for realization of 1-D DWT. [Longa *et al*. (2008), Mohanty and Meher (2009)]. Cao *et al*. (2006) have derived a DA structure for 1-D DWT based on new DA algorithm suggested by [Pan *et al*. (1999)] and subsequently applied compression technique to reduce adder complexity of the structure. Longa *et al*. (2008) have suggested a LUT-less DA design for implementation of DWT core. They have implemented DA-LUT using adders and multiplexers (MUXes). However, the adder complexity of structure is significantly higher than the previous multiplier less structures. Few more DA-based designs also have been proposed in [Mohanty and Meher (2009), Mahajan and Mohanty (2013)] for 9/7 wavelet filters using the symmetric property and carry-save arithmetic.

Table 2.1 Summary of DWT designs proposed in technical literature

Designs	**Computation Scheme**	**Design type**	**Design Issue**
Lewis and Knowles (1991)	Convolution	PA	Systolization
Parhi and Nishitani (1993)	Convolution	PA	Systolization
Grzeszazak *et al.* (1996)	Convolution	RPA	Systolization with less delay
Denk and Parhi (1998)	Convolution	PA	Systolization with less delay
Wu and Chen(1999)	Convolution	PA	Improve utilization efficiency
Marino *et al* (2000)	Convolution	RPA	Improve utilization efficiency
Wu and Chen (2001)	Convolution	Folded	Simplified design with 100% utilization efficiency
Jou *et al.* (2001)	Lifting	Folded	Reduce arithmetic complexity
Huang *et al*. (2002)	Flipping	Folded	Reduce the critical path
Week and Bayoumi (2003)	Lifting	Folded	Systolization
Liao *et al*. (2004)	Lifting	Folded /RPA	Increase throughput rate
Die *et al*. (2004)	Convolution	Folded	Multi-dimensional scalable design
Barua *et al*. (2005)	Lifting	Folded	Reduce critical path with less overhead
Xiong *et al*. (2006)	Lifting	Hybrid (PA+RPA)	Improve utilization efficiency
Xiong *et.al* (2007)	Lifting	Folded	Generalized design and less critical-path delay
Seo and Kim(2007)	Lifting	Folded	Critical path delay optimization with less overhead

Cheng *et al*. (2007)	Lifting	Folded	On chip memory optimization
Cheng and Parhi (2008)	Convolution	RPA (block-based)	Reduction in arithmetic complexity
Lai *et al*. (2009)	Flipping	Folded	Critical-path delay optimization with less overhead
Tian *et al*. (2011)	Lifting	Folded (block-based)	Throughput-rate
Mohanty and Meher (2011)	Lifting	Parallel (PA+RPA)	Improve memory and resource utilization
Mohanty *et.al.* (2012)	Lifting	Folded (block-based)	Improve memory utilization
Zhang *et.al.* (2012)	Lifting	Folded	Critical-path delay optimization with less overhead
Mohanty and Meher (2013)	Convolution	Parallel (PA+RPA)	On-chip memory optimization, and resource utilization
Hu and Jong (2013)	Flipping	Parallel (PA)	On-chip memory optimization, and resource utilization
Darji and Limaye(2014)	Lifting	Folded	On-chip memory and critical-path delay optimization with less overhead
Darji *et.al.* (2014)	Flipping	Folded	On-chip memory and critical-path delay optimization with less overhead
Bhanu and Chilambuchelvan(2014)	Lifting	Folded	Reduce computation time
Hu and Jong (2015)	Flipping	Parallel(PA)	On-chip memory optimization
Maye and Srinivasulu (2015)	Lifting	Folded	On-chip memory optimization
Wang and Choy (2016)	Lifting	Folded	On-chip memory and critical path optimization
Zhang *et.al.* (2016)	Lifting	Parallel	Increase throughput-rate and hardware utilization

Table 2.1 summarizes the issue consider while designing the computing structure for 2-D DWT. Although the systolic design offers a computing structure with high degree of regularity and modularity, but it involves a large number of pipeline registers which contribute significantly to the area complexity and power consumption. The folded scheme along with lifting and flipping computation helps to reduces the multiplier, adder, on-chip memory complexity and critical-path delay of 2-D DWT designs, the critical-path delay optimization with less overhead cost becomes the design issue for many designs. The frame-buffer of folded structure becomes an issue for single-chip realization of complete 2-D DWT structure. Parallel designs have been proposed to avoid the frame-buffer and improve the memory utilization of multi-level 2-D DWT structure. Different types of mapping schemes have been proposed to reduce the on-chip memory of parallel structures which is higher than the folded structure. It has been observed that DWT designs involve data-selectors (multiplexors and de-multiplexors) apart from multiplier, adder and memory elements. DWT structures use data-selectors for time-multiplexing data sequences to improve resource utilization of the hardware design while performing down-sample filter computation. Data-selector complexity depends on block size (throughput rate) of folded and RPA designs. In case of parallel designs, data-selector complexity depends on mapping algorithm, input block-size and decomposition levels. Due to throughput-scalability feature the full-parallel design of 2-D DWT involves a large input block-size for higher decomposition-level. The data-selector complexity of such parallel designs is relatively large compared to the folded and RPA designs and that affect the area-delay efficiency of the design substantially. However, the data-selector complexity is overlooked in the existing parallel designs. Also, it is observed that the arithmetic unit of parallel design of large block size contributes almost comparable amount area as the on-chip memory unit. But, in the existing parallel structures the design effort is focused on the on-chip memory optimization. Since the parallel structure of large block size involves hundreds of multipliers, use of an optimized hard multiplier could improve the area-delay efficiency substantially. Although, few memory-less designs have been proposed in the technical literature for low-complexity realization of DWT, but these designs use convolution scheme. Look-up-table based multiplier design could be a better choice for the parallel designs which is yet to be explored.

2.5 PROBLEM FORMULATION

(i) In parallel design, data-blocks are reordered and time-multiplexed at different stages of DWT computation. The format of data-reordering depends on the input data-access scheme used by the parallel design. The input-output data-flow of different levels of 2-D DWT computation needs a detail study to find the reordering mechanism of intermediate data-blocks. Based on this finding, a different data-access scheme needs to be formulated to avoid data-ordering in parallel computation of 2-D DWT. The lifting 2-D DWT algorithm needs to be presented in a modified form to facilitate algorithm mapping into the parallel architecture according to the new data access scheme. An appropriate architecture need to be designed for parallel computation of lifting 2-D DWT without using data-selectors. These issues will be addressed in Chapter 3.

(ii) Radix-4 Booth algorithm is popularly used for implementation of sign multiplication in hardware design. The higher radix Booth multiplication algorithms reduces the number of partial product rows of the partial product array (PPA) which helps to reduce the adder unit complexity, but higher radix Booth algorithm increases the complexity of partial product generator (PPG) and partial product selector (PPS). Consequently, the area-delay efficiency of higher radix Booth multiplier designs appears not better than the radix-4 Booth multiplier design. However, area-delay efficiency of radix-8 Booth multiplier design is closest to radix-4 Booth multiplier design. With appropriate algorithm modification and logic optimization, the radix-8 Booth multiplier design could be made hardware efficient than the radix-4 Booth multiplier design for specific bit-width. Chapter 4 addresses this issue.

(iii) A parallel design of lifting 2-D DWT involves large number of constant multipliers. Many of these constant multipliers share a common input operand. This is an interesting feature of parallel design and could be exploited using higher radix Booth multiplier which is currently missing in the literature. The radix-4 Booth multiplier does not provides any area saving in 2-D DWT design due to common input operand. Chapter 5 explores the possibility of area saving in parallel designs of lifting 2-D DWT using radix-8 Booth multiplier design.

(iv) Lifting DWT algorithm does not support the multiplier-less designs such as MCM, CSE and DA schemes. A look-up-table (LUT) based multiplier design could be considered for lifting

DWT to have a multiplier-less design. Several design schemes have been proposed in the technical literature for efficient realization of LUT multiplier. However, these LUT-multiplier designs are proposed for multiplication of unsigned numbers. The lifting DWT based on (9/7)-wavelet filters involve sign multiplication. Chapter 6 discusses the issues related to sign multiplication using LUT and explore the possibility of using LUT multiplier in parallel design of lifting 2-D DWT.