Fast Implementation of Lifting Based DWT Architecture for Image Compression

Dr. M. Nagabushanam[1] and S. Ramachandran[2]

[1] Anna University, Coimbatore, India.

Received: 6 February 2012 Accepted: 1 March 2012 Published: 15 March 2012

Abstract

Technological growth in semiconductor industry have led to unprecedented demand for faster, area efficient and low power VLSI circuits for complex image processing applications. DWT-IDWT is one of the most popular IP that is used for image transformation. In this work, a high speed, low power DWT/IDWT architecture is designed and implemented on ASIC using 130nm Technology. 2D DWT architecture based on lifting scheme architecture uses multipliers and adders, thus consuming power. This paper addresses power reduction in multiplier by proposing a modified algorithm for BZFAD multiplier. The proposed BZFAD multiplier is 65

Index terms— DWT, Image compression, BZFAD multiplier, FPGA, Lifting scheme.

1 Introduction

he wavelet transformation is a widely used technique for image processing applications. Unlike traditional transforms such as the Fast Fourier Transform (FFT) and Discrete Cosine Transform (DCT), the Discrete Wavelet Transform (DWT) holds both time and frequency information, based on a multiresolution analysis framework. This facilitates improved quality of reconstructed picture for the same compression than is possible by other transforms. In order to implement real time Codecs based on DWT, it needs to be targeted on a fast device. Field Programmable Gate Array (FPGA) implementation of DWT results in higher processing speed and lower costs when compared to other implementations such as PCs, ARM processors, DSPs etc. The Discrete wavelet transform is therefore increasingly used for image coding [1][2][3][4]. This is because the DWT can decompose the signals into different sub-bands with both time and frequency information and facilitate to arrive a high compression ratio [5]. It supports features like progressive image transmission (by quality, by resolution), ease of compressed image manipulation, region of interest coding, etc.

The JPEG 2000 incorporates the DWT into its standard [6]. Recently several VLSI architectures have been proposed to realize single chip designs for DWT [7][8][9][10]. Traditionally, such algorithms were implemented using programmable DSP chips for low-rate applications or VLSI application specific integrated circuits (ASICs) for higher rates. To perform the convolution, we require a fast multiplier which is crucial in making the operations efficient.

2 II.

Lifting based dwt scheme decomposition. This process is continued as per the design requirements till the requisite quality is obtained. Every stage of DWT requires LPF and HPF filters with down sampling by 2. Lifting based DWT computation is widely being adopted for image decomposition. In this work, we propose a modified architecture based on BZFAD [11] multiplier to realize the lifting based DWT.

Lifting scheme is one of the techniques that is used to realize DWT architecture. Lifting scheme is used in order to reduce the no of operations to be performed to half and filters can be decomposed into steps in lifting scheme. The memory required and also computation is less in case of lifting scheme. The implementation of the algorithm is fast and inverse transform is also simple in this method. The Fig. 2.shows the block diagram for lifting scheme [12]. The z -1 blocks are for delay, ?, ?, ?, ?, ? are the lifting coefficients and the shaded blocks are registers.9/7 filter has been used for implementation which requires four steps for lifting and one step for scaling. The input signal is x i is split into two parts even part x 2i and odd part x 2i+1 then the first step of lifting is performed given by the equations [13]:d i 1 = ? (x 2i + x 2i+2) + x 2i+1**(1)**a i 1 = ? (d i 1 + d i 1 -1) + x 2i**(2)**

The first equation is predict P1 and second equation is update U1.Then the second lifting step is performed which gives [13]:d i 2 = ? (a i 1 + a i 1 +1) + d i 1**(3)**a i 2 = ? (d i 2 + d i 2 -1) + a i 1**(4)**

The third equation is predict P2 and fourth equation is update U2.Then scaling is performed and the following equations are obtained [13]:a i = ? a i 2 = G 1 (**5**)d i = d i 2 ? ? = G 2**(6)**

The equations 5 and 6 are scale G 1 and G 2 respectively. The predict step helps determine the correlation between the sets of data and predicts even data samples from odd. These samples are used in the update step for updating the present phase. Some of the properties of the original input data can be maintained in the reduced set also by construction of a new operator using the update step. The lifting coefficients have constant values of -1.58613, -0.0529, 0.882911, 0.44350, -1.1496 for ?, ?, ?, ?, ? respectively. By observation of the above equations, computing the final coefficients requires 6 steps. Data travels in sequence from stage 1 to stage 6, this introduces a delay of 6 stages. To speed up the process of computation, modified lifting scheme is proposed and realized.

3 III.

4 Arithmetic building blocks for lifting scheme implementation

High-speed multiplication has always been a fundamental requirement of high performance systems. Multiplier structure is one of the processing element consumes the maximum area and power and also constitutes delay. Therefore there is a need for highspeed architectures for N-bit multipliers with optimized area, speed and power. Multipliers are made up of adders, to reduce the Partial Product Reduction logic delay and regularize the layout. To improve regularity and compact layout, regularly structured tree with recurring blocks and rectangular-styled tree by folding are proposed at the expense of more complicated interconnects [14]. The present work focuses on multiplier design for low power applications such as DWT by rapidly reducing the partial product rows by identifying the critical paths and signal races in the multiplier. In other words, the goals have been to optimize the speed, area and power of the multiplier that form the major block in lifting based DWT.

5 a) Shift and Add Multiplier

In shift and add based multiplier logic, the multiplicand (A) is multiplied by multiplier (B). If the register A and B storing multiplicand and multiplier respectively is of N bit, the shift and add multiplier logic requires two N bit registers, and an N bit adder and N+1 accumulator. It also requires a N-bit counter to control the number of addition operation. In shift and add logic, the LSB bit of multiplier is checked for 1 or 0, if the LSB bit is 0, then the accumulator is shifted right by 1-position. If the LSB bit is 1 then the multiplicand is added with the accumulator content and the accumulator is shifted right by one bit position. The counter is decremented for every operation; the addition is performed until the counter is set to zero, which is indicated by the signal Ready. The multiplied product available in the accumulator of N clock cycles is the final output. Figure 3

6 Bz-fad multiplier

As discussed in shift and add logic, if the LSB position is 1 then the accumulator is added with the multiplicand. If the accumulator contains more number of 1s, the adder has to add the 1 and this triggers the Full adder block within the adder. As we know that the power dissipation is due to switching activity of input lines, when ever the input or output changes, the power is switched from Vdd to Vss, thus contributing power dissipation. In order to reduce power dissipation, it is requried to reduce switching activity in the I/O lines. BZ-FAD [23] logic based multiplier reduces the switching activity and thus reduces power dissipation. In shift and logic for every operation the counter keeps track of number of cycles and thus controls the multiplication operation. In a binary counter, we know that the output bit change occurs in more than one bit, for example if the counter output is 2 and is changing to 3, there are two bit change occurring. This causes switching activity, and thus can be reduced by replacing the binary counter by ring counter. In a ring counter, at any given point of time only one bit change occurs, thus reducing switching activity and power dissipation. Another major source of power dissipation in shift and add logic is, for every bit 0 of the multiplier a shift operation is performed, thus all the bits in the accumulator are shifted by one bit position, this also introduces switching and thus power dissipation. In BZ-FAD logic, if the LSB bit is 0, then the shift operation is bypassed and a zero is introduced at the MSB, thus there is no shifting of accumulator content. In other words, if the LSB is zero, the accumulator is directly fed into the adder and there is no addition, but a zero is introduced by the control logic which is like right shift operation. The architecture of this multiplier is shown in Figure **??**. Fig. **??** : Low power multiplier architecture [16] As the BZFAD, the control activity of ring counter, latch and bypass logic is realized using NMOS transistors, this introduces delay. The parasitic capacitance of NMOS transistors also increases the load capacitance and thus increases power dissipation. In order to reduce power dissipation we have replaced the transistor logic by

MUX logic that have been designed to have ideal fanin and fanout capacitances. With MUX based logic the control signals can be suitably controlled to reduces switching activity as they are enabled only when required, based on the inputs derived from ring counter. However, the design requires more number of transistors and thus increases the chip area. We have also used the ripple carry adder which has the least average transition per addition among the look ahead, carry skip, carry-select and conditional sum adders to reduce power dissipation. Various multipliers are modeled in HDL and are analyzed for their performances and the results are tabulated for comparison. Next section discusses the comparison results of multiplier algorithms.

7 a) Comparison of Results

In this section, comparison of power, area for different types of multiplier with modified multiplier (BZ-FAD) is discussed. The results reveal that the modified BZ-FAD multiplier may be considered as a very lowpower, yet highly area efficient multiplier. linearly with the input data width. This leads to a small increase in the leakage power which, as the results reveal, is less than the overall power reduction. The leakage power of the 8-bit BZFAD architecture is about 11% more than that of the conventional architecture but the contribution of the leakage power in these multipliers is less than 3% of the total power for the technology used in this work. Finally, note that since the critical paths for both architectures are the same neither of the two architectures has a speed advantage over the other.

8 Discrete wavelet transform and Inverse Discrete wavelet transform implementation

The discrete wavelet transform (DWT) is being increasingly used for image coding. This is due to the fact that DWT supports features like progressive image transmission (by quality, by resolution), ease of compressed image manipulation, region of interest coding, etc. DWT has traditionally been implemented by convolution. Such an implementation demands both a large number of computations and a large storage features that are not desirable for either high-speed or low-power applications. Recently, a lifting-based scheme that often requires far fewer computations has been proposed for the DWT [20,21,22]. The main feature of the lifting based DWT scheme is to break up the high pass and low pass filters into a sequence of upper and lower triangular matrices and convert the implementation into banded matrix multiplications. Such a scheme has several advantages, including ”in-place” computation of the DWT, integer-to-integer wavelet transform (IWT), symmetric forward and inverse transform, etc. Therefore, it comes as no surprise that lifting has been chosen in the upcoming.

The proposed architecture computes multilevel DWT for both the forward and the inverse transforms one level at a time, in a row-column fashion. There are two row processors to compute along the rows and two column processors to compute along the columns. While this arrangement is suitable or filters that require two banded-matrix multiplications filters that require four banded-matrix multiplications require all four processors to compute along the rows or along the columns. The outputs generated by the row and column processors (that are used for further computations) are stored in memory modules.

The memory modules are divided into multiple banks to accommodate high computational bandwidth requirements. The proposed architecture is an extension of the architecture for the forward transform that was presented. A number of architectures have been proposed for calculation of the convolution-based DWT. The architectures are mostly folded and can be broadly classified into serial architectures (where the inputs are supplied to the filters in a serial manner) and parallel architectures (where the inputs are supplied to the filters in a parallel manner).

Recently, a methodology for implementing lifting-based DWT thatreduces the memoryrequirements and communication between the processors, when the image isbroken up into blocks. For a system that consists of the lifting-based DWT transform followed by an embedded zero-tree algorithm, a new interleaving scheme that reduces the number of memory accesses has been proposed. Finally, a lifting-based DWT architecture capable of performing filters with one lifting step, i.e., one predict and one update step. The outputs are generated in an interleaved fashion. The Discrete wavelet transforms and inverse discrete wavelet transform operates at a maximum clock frequency of 200MHz. the discrete wavelet transforms and inverse discrete wavelet transform is synthesized by using design compiler. The design of DWT and IDWT is checked design for testability. Every time checked(D D D D) F 2012 Year h(i) =x(2i+1)+?(x(2i)+x(2i+2)) l(i)=x(2i)+?(h(i)+h(i-1)) hh(i, j) = h(2i +1, j) +? (h(2i, j) + h(2i + 2, j)) hl(i, j) = h(2i, j) + ? (hh(i, j) + hh(i ?1, j)) lh(i, j) = l(2i +1, j) +? (l(2i, j) + l(2i + 2, j))

ll(i, j) = l(2i, j) + ? (lh(i, j) + lh(i ?1, j)) l(2i,j)= ll(i,j)-?(lh(i,j)+lh(i-1,j)) l(2i+1,j)= lh(i,j)-?(L(2i,j)+l(2i+2,j)) h(2i,j)= hl(i,j)-?(hh(i,j)+hh(i-1,j)) h(2i+1,j)= hh(i,j)-?(h(2i,j)+h(2i+2,j))

x(i,2j) = l(i,j)-?(h(i,j)+h(i,j-1))

x(i,2j+1) = h(i,j)-?(x(i,2j)+x(i,2j+2)) timing reports and the power report is taken from the primetime. The architectures for DWT and IDWT perform compression and decompression in (4N 2 (1?4? j) + 9N)/6 computation time. The total power consumption of the DWT/IDWT processor is ~0.367mW. The area of the designed architecture in 0.13 micron technology is 112 X 114 um square, and the frequency of operation is 200 MHz for discrete wavelet transform.

9 VI.

10 Conclusion

In this work low-power architecture for shift-andadd multipliers is proposed and implemented. The conventional architecture has been modified by removing the shift operation of the B register (in A × B), direct feeding of A to the adder, bypassing the adder whenever possible, use of a ring counter instead of the binary counter, and removal of the partial product shifter. The BZ-FAD multiplier is further modified using multiplexers and XOR gates, the modified multiplier is modeled and implemented using 130nm technology. The modified multiplier is used in constructing lifting based DWT/IDWT architecture. The DWT/IDWT architecture is modeled and synthesized using TSMC libraries. The BZ-FAD multiplier based DWT/IDWT architecture reduces power dissipation by 30% and operates at 200 MHz. The adders in the lifting based DWT/IDWT can be further improved by replacing the adders by low power adders.

1

Figure 1: FigFigure 1 :

2

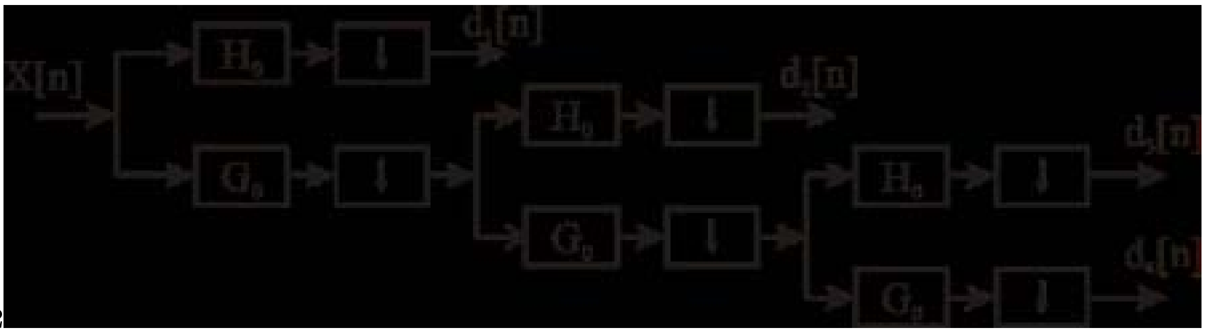

Figure 2: Figure 2 :

3

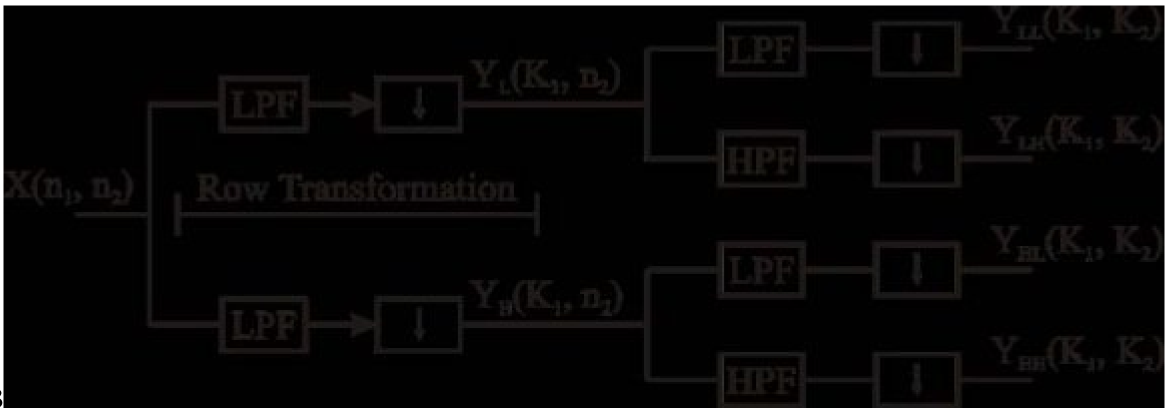

Figure 3: Fig. 3 :

5

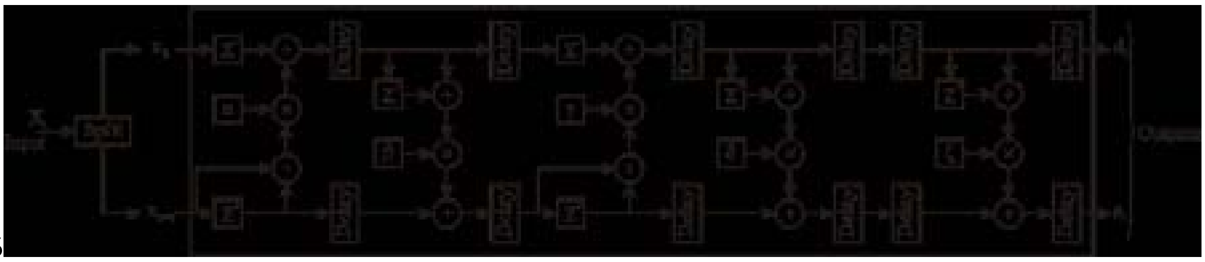

Figure 4: Fig. 5 :

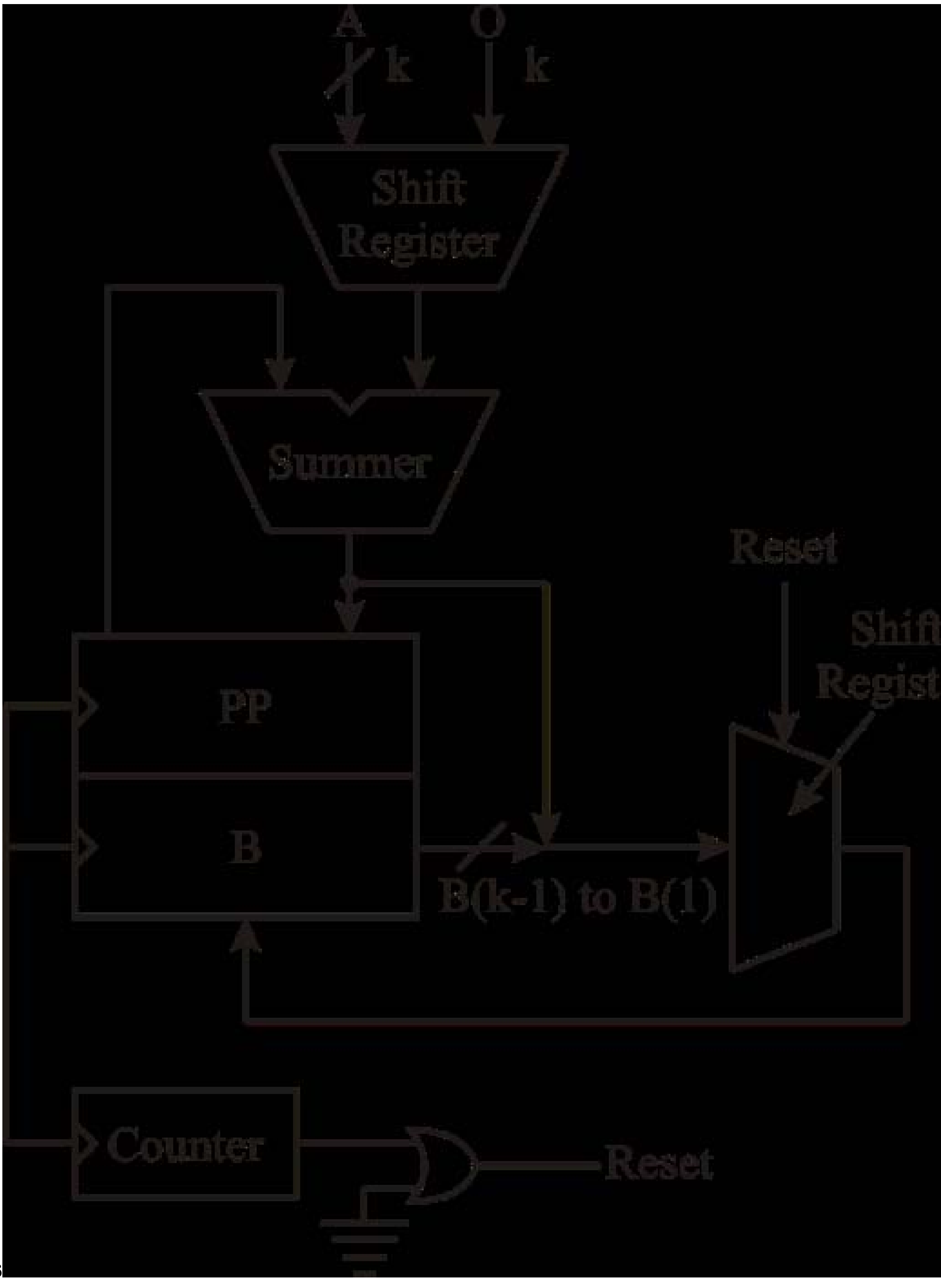

Figure 5: Fig. 6 :

2

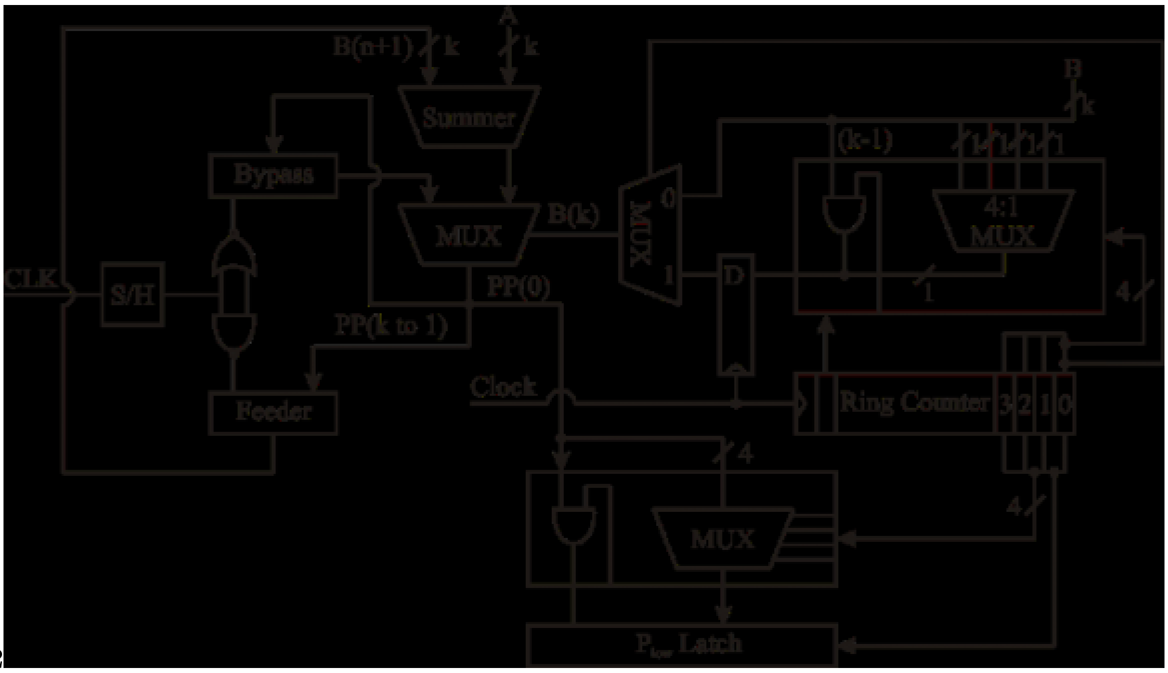

Figure 6: Figure 2 .

1

Multipliers	other multipliers Total Dynamic power (w)	Cell Internal Power (μw)	Net Switching Power	Cell Leakage power (μw)
Modified BZ-FAD Multiplier	126	91.02	21.2	13.78
Shift and Add Multiplier	194	166.9	15.2	11.9
Booth Multiplier	379.12	295.62	62.2	21.3
Array Multiplier	231.5	145.4	66.3	19.8
Wallace Tree Multiplier	289.9	195.9	76.9	17.1

Figure 7: Table 1 :

2

Multipliers	Total Cell Area (µm2)	Number of Ports	Number of nets	Number of cells	Number of References
BZ-FAD Multiplier	2479.9 0	35	133	74	43
Shift and Add Multiplier	1726.2 5	35	99	43	12
Booth Multiplier	4459.0 6	34	233	163	32
Array Multiplier	3213.2 7	34	228	156	66
Wallace Tree Multiplier	3476.2 7	34	241	160	67

Figure 8: Table 2 :

[Kang and Gaudiot (2004)] 'A Fast and Well structured Multiplier'. J-Y Kang , J-L Gaudiot . *EUROMICRO Symp. Digital System Design*, Aug 2004. p. .

[Kuan et al. (2007)] 'A Low-Power Multiplier with the Spurious Power Suppression Technique'. Hung Kuan , Yuan-Sun Chen , Chu . *IEEE Trans. On Very Large Scale Integration (VLSI) Systems* July 2007. 15 (7) p. .

[Movva and Srinivasan (2003)] 'A novel architecture for lifting-based discrete wavelet transform for JPEG2000 standard suitable for VLSI Implementation" VLSI Design'. S Movva , S Srinivasan . *Proceedings. 16 th International Conference On 4-8*, (16 th International Conference On 4-8Page(s) 2003. Jan. 2003. p. .

[Dusansuvakovic and Andre ()] 'A Pipelined Multiply-Accumulate Unit Design for Energy Recovery DSP Systems'. C Dusansuvakovic , Salama Andre . *IEEE International Symposium on Circuits and Systems*, May 28-31, 2000.

[Kang and Gaudiot (2006)] 'A Simple High-Speed Multiplier Design'. Jung-Yup Kang , Jean-Luc Gaudiot . *IEEE Transactions on computers* October 2006. 55 (10) p. .

[Wallace ()] 'A Suggestion for a Fast Multiplier'. C S Wallace . *IEEE Trans. computers* 1964. 13 (2) p. .

[Andra and Chaitalichakrabarti (2002)] 'A VLSI Architecture for Lifting-Based Forward and Inverse Wavelet Transform'. Kishore Andra , Tinkuacharya Chaitalichakrabarti . *IEEE Transaction on signal processing* April 2002. 50 (4) p. .

[Motra et al. (2003)] 'An efficient hardware implementation of DWT and IDWT'. A S Motra , P K Bora , I Chakrabarti . *Conference on Convergent Technologies for Asia-Pacific Region*, October 2003. p. .

[Mottaghi-Dastjerdi et al. (2009)] 'BZ-FAD: A Low-Power Low-Area Multiplier Based on Shift-and-Add Architecture'. M Mottaghi-Dastjerdi , A Afzali-Kusha , M Pedram . *IEEE Transactions on Very large Scale Integration (VLSI)systems*, feb 2009. 17.

[Vojin et al. (2005)] 'Comparison of high-performance VLSI adders in the energy-delay space'. G Vojin , Bart R Oklobzija , Zaydel , Q Hoang , Sanu Dao , Ram Mathew , Krishnamurthy . *IEEE Trans. VLSI Systems* June 2005. 13 (6) p. .

[Nagabushanam et al. ()] 'Design and FPGA implementation of Modified Distributive arithmetic based DWT-IDWT processor for image compression'. M Nagabushanam , Cyril Prasanna Raj , S Ramachandran . *International Conference on Communication and Signal Processing*, (February, NIT, Calicut, India) 2011. p. 69.

[Nagabushanam et al. ()] 'Design and implementation of parallel and pipelined Distributive arithmetic based discrete wavelet transform IP core'. M Nagabushanam , Cyril Prasanna Raj , S Ramachandran . *EJSR International Journal* 2009. 35 (3) p. .

[Chang and Li (2001)] 'Design of highly efficient VLSI architectures for 2-D DWT and 2-D IDWT'. Yun-Nan Chang , Yan-Sheng Li . *IEEE Workshop on Signal Processing Systems* September 2001. p. .

[Tsai and Bdti] *Designing Low-Power Signal Processing Systems*, Mel Tsai , Bdti . `http://www.dspdesignline.com/showArticle.jhtml?articleID=187002923`

[Borgio et al. (2006)] 'Hardware DWT accelerator for MultiProcessor System-on-Chip on FPGA'. Simone Borgio , Davidebosisio , Fabrizioferrandi , Marco D Matteomonchiero , Donatella Santambrogio , Antoninotumeo Sciuto . *Embedded Computer Systems: Architectures, Modeling and Simulation*, 2006. 2006. July 2006. p. .

[Huang and Ercegovac (2005)] 'Highperformance Low-power Left-to-Right Array Multiplier Design'. Zhijun Huang , Milos D Ercegovac . *IEEE Transactions on computers* Mar 2005. 54 (3) p. .

[Taubman and Marcellin ()] *JPEG 2000 -Image compression, fundamentals, standards and practice*, David S Taubman , Michael W Marcellin . 2002. Kluwer academic publishers. (Second Edition)

[Baloch et al. ()] 'Low power domain-specific reconfigurable array for discrete wavelet transforms targeting multimedia applications'. S Baloch , I Ahmed , T Arslan , A Stoica . *International Conference on Field Programmable Logic and Applications*, 2005. p. . (International Conference on Field Programmable Logic and Applications)

[Tze-Yun ()] 'Low-power and high-performance 2-D DWT and IDWT architectures based on 4-tap Daubechies filters'. Tze-Yun . *Proceedings of the 7th WSEAS International Conference on Multimedia Systems and Signal Processing*, (the 7th WSEAS International Conference on Multimedia Systems and Signal Processing-Hangzhou, China) 2007. p. .

[Sung et al. (2006)] 'Low-Power Multiplierless 2-D DWT and IDWT Architectures Using 4-tap Daubechies Filters'. Tze-Yun Sung , Hsi-Chin Hsin Yaw-Shih , Chun-Wang Shieh , Yu . *Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies*, December 2006.

[Nagabushanam et al.] 'ModifiedVLSI implementation of DA-DWT for image compression'. M Nagabushanam , Cyril Prasanna Raj , S Ramachandran . *International Journal of Signal and Imaging Systems Engineering* x (x) .

[Keshab et al. (1993)] ‘VLSI architectures for discrete wavelet transforms’. K Keshab , Takao Parhi , Nishitani . *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, June 1993. 1 p. .

[Abdullah et al. (2004)] ‘VLSI Implementation of Discrete Wavelet Transform (DWT) for Image Compression’. A L Abdullah , Md Muhit , Masuri Islam , Othman . *2nd International Conference on Autonomous Robots and Agents*, December 2004. p. .