2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems

An Efficient VLSI Architecture for Discrete Hadamard Transform

Mohamed Asan Basiri M,
Department of Computer Science and Engineering, Indian Institute of Information Technology Design and Manufacturing Kancheepuram, Chennai 600127.
Email: asanbasiri@gmail.com

Noor Mahammad Sk,
Department of Computer Science and Engineering, Indian Institute of Information Technology Design and Manufacturing Kancheepuram, Chennai 600127.
Email: noorse@gmail.com

***Abstract*—This paper proposes an efficient VLSI architecture for discrete Hadamard transform, which is used in real time digital signal processing applications like image coding, MPEG, and CDMA etc. The proposed N-point Hadamard transform architecture consists of signed carry save adder tree. So the depth of the architecture falls within the bounds of $O(log_2\ N)$. The same proposed architecture is implemented for 2D-discrete Hadamard transform, where the hardware utilization is full. In other words, row/column processing of 2D-transform is carried with same 1D-hardware. The performance results show that the proposed architecture gives better performance compared with existing architectures using $45\ nm$ CMOS TSMC library. The proposed 8-point 1D-DHT achieves an improvement of 55.14% and 42.07% in worst path delay over conventional and linear systolic array based 1D-DHT architectures respectively. Similarly proposed 8×8-point 2D-DHT achieves an improvement of 50.7%, 21.7%, and 51.1% in worst path delay over conventional, linear systolic array, and distributed arithmetic based 2D-DHT architectures respectively.**

***Keywords*-Distributed arithmetic; DSP processor; Hadamard Transform;**

I. Introduction

Digital signal processors (DSPs) are essential for real-time processing of real-world digitized data to perform high-speed numeric calculations used for a broad range of applications from basic consumer electronics to sophisticated industrial instrumentation. The discrete transform [1] is used to change the representation of a signal from one domain to another for reducing the complexity of a particular digital signal processing application. The discrete Hadamard transform (DHT) is a real discrete orthogonal transform (DOT), which can be widely used in CDMA [2] and MPEG [3] applications. In image processing, DHT is used for an efficient implementation of image compression and coding [4]. The equation (1) represents the 1D-forward discrete Hadamard transform of N-point input signal sample values (X), where Y is represented as N-point output signal sample values and H_N is $N \times N$-point discrete Hadamard co-efficient matrix. Here the given input vector is multiplied with matrix (H_N).

$$Y_N = H_N X_N \tag{1}$$

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2}$$

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{3}$$

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} \tag{4}$$

$$H_N = \begin{bmatrix} H_{\frac{N}{2}} & H_{\frac{N}{2}} \\ H_{\frac{N}{2}} & -H_{\frac{N}{2}} \end{bmatrix} \tag{5}$$

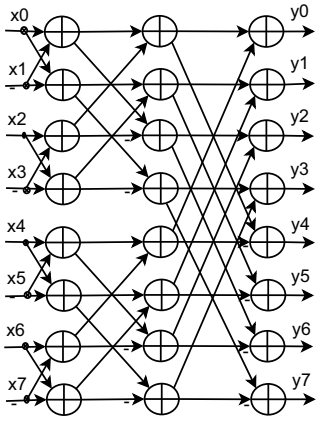

Figure 1. Conventional architecture of 8-point discrete Hadamard transform

The equations (2), (3), and (4) represent 2, 4, and 8-point discrete Hadamard co-efficient matrices respectively. The general N-point discrete Hadamard co-efficient matrix can be derived from $\frac{N}{2}$-point Hadamard co-efficient matrix, which is represented in equation (5). Figure 1 represents 1D-conventional architecture [5] of 8-point discrete Hadamard transform, where x and y are input and output signal values respectively. Here $log_2 N$ stages of addition will be involved and each stage consists of N numbers of signed adders. Therefore, total number of signed adders in N-point 1D-DHT architecture is $Nlog_2 N$.

Another VLSI architecture for 1D-DHT is proposed in [6], where N-point DHT can be obtained by using carry save addition based 4-point DHTs. Here, the given N-point 1D-DHT is started with signed adders and ended with 4-point

978-1-4673-8700-2/16 $31.00 © 2016 IEEE
DOI 10.1109/VLSID.2016.63

carry save adders. Since the carry save addition used in the architecture, delay for [6] is less than [5]. Therefore, [6] is faster than [5]. Distributed arithmetic based 1D-DHT architectures are proposed in [7] and [8], where look up table based matrix-vector multiplication is proposed. The 2D-DHT architectures are explained in [10] and [11]. Usually, 2D transform can be implemented by using 1D architecture. If the input signal sample values are represented as $N \times N$ matrix, then N numbers of N-point 1D-DHTs will be performed during row process. Here, 1D-DHT for each row of input signal matrix will be carried. In the next step, transpose will be taken for the matrix obtained from row process. In the last step (column process), N numbers of N-point 1D-DHTs will be performed for each row of transposed row process matrix.

A. Contribution of this paper

The objective of this work is to improve the performance of DHT architecture for DSP applications. In general, performance of a circuit depends on circuit delay, circuit area, and net power requirement. The circuit delay is defined as the path from input to output with largest (worst path) delay in the circuit. The careful optimization in these parameters will ensure the highest performance. In this work, circuit delay is considered as the primary objective. The proposed N-point Hadamard transform architecture is designed with signed carry save adder (CSA) tree. So the depth of the architecture falls within the bounds of $O(log_2\ N)$. The same proposed architecture is implemented for 2D-discrete Hadamard transform, where row/column processing is carried with same hardware. Therefore, the hardware utilization is full. The performance results show that the proposed architecture gives better performance compared with existing architectures.

The rest of the paper is organized as, section II explains the motivation for the proposed design and section III states the proposed architecture for 1D and 2D-DHTs. Design modeling, implementation, and results are stated in section IV, followed by a section V as conclusion.

II. Motivation for the Proposed Design

In the conventional design of N-point 1D-DHT [5], worst (critical) path includes $2(log_2 N)$ number of CLAs. Since signed addition is considered, each stage includes 2 CLAs (one for taking 2's complement according to hadamard co-efficient and an another for addition of 2 input values, which is shown in Figure 2. The total number of signed addition stages in conventional design is $log_2 N$. Therefore, the critical path of conventional design contains $2(log_2 N)$ number of recursive doubling based carry look ahead adders (CLAs).

In case of linear systolic array based architecture [6], last 2 signed addition stages of conventional design are replaced by 4-point signed CSA. Here 4-point signed CSA

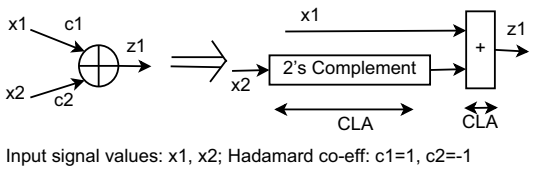

Figure 2. Signed addition of discrete Hadamard transform

includes 2 carry save stages and one CLA. Therefore, the total number of CLA and CSA stages in critical path of [6] is $2(log_2 N) - 2$ and 2 respectively. In case of distributed arithmetic based architecture [8], first and last stages consist of signed additions. The middle (second) stage includes look up table, where look up table contents are filled with signed additions. Therefore, the critical path in [8] includes, 4 CLAs and look up table ($T(LUT)$). Since the N-point discrete Hadamard co-efficients belong to $\{-1,\ 0,\ 1\}$, $T(LUT)$ represents the time complexity for adding $\frac{N}{2}$ numbers of -1 and/or $+1$ and/or 0. If the input sample values are n-bits wide then, [8] requires n-clock cycles to produce final outputs. This is the main draw back with distributed arithmetic based circuit design.

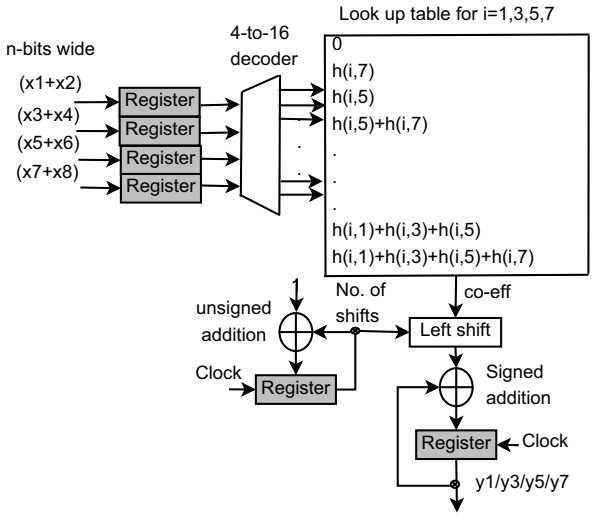

Figure 3. Distributed arithmetic based 8-point 1D-DHT

Figure 3 shows the distributed arithmetic based 8-point 1D-DHT architecture, where $x1$, $x2$, ...$x8$ are n-bit input values and $y1$, $y2$, ...$y8$ are n-bit output values. The discrete Hadamard co-efficients are represented as $h(i,j)$, where i and j will be varied from 0 to 7. Here least significant bits (lsb) of $(x1+x2)$, $(x3+x4)$, $(x5+x6)$,and $(x7+x8)$ are sent to look up table, where the corresponding signed added value of Hadamard co-efficient is obtained. In final stage, $co\text{-}eff$ value obtained from look up table is left shifted and the shifted value is added with previous result. During each clock cycle, $(x1+x2)$, $(x3+x4)$, $(x5+x6)$, and $(x7+x8)$

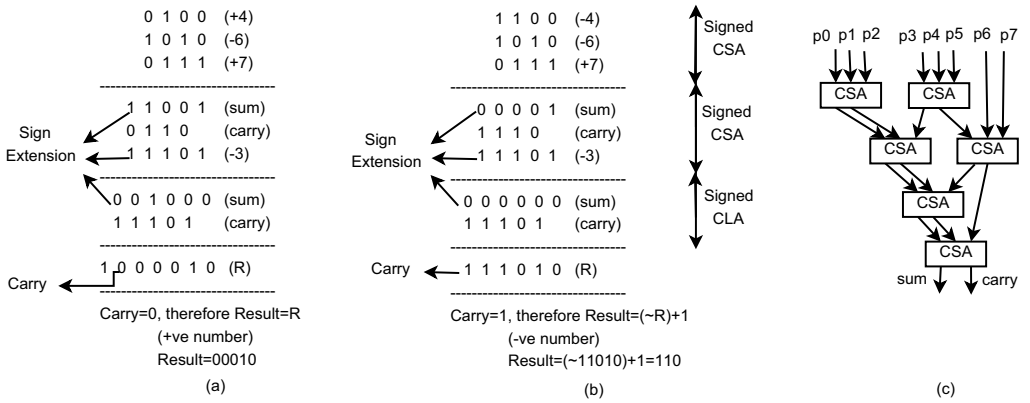

Figure 4. Signed carry save addition (a) Example: 4-6+7-3 (b) Example: -4-6+7-3 (c) 8-point carry save reduction tree

are one bit position right shifted. Therefore, it requires n clock cycles to produce the 1D-DHT outputs. The 4-to-16 decoder is used to obtain the corresponding $co\text{-}eff$ from look up table. The number of left shifts for $co\text{-}eff$ will be incremented during each clock cycle. This architecture can produce odd output values $y1$, $y3$, $y5$, and $y7$. Similarly even output values $y2$, $y4$, $y6$, and $y8$ can be obtained by using another architecture with $(x1 - x2)$, $(x3 - x4)$, $(x5 - x6)$, and $(x7 - x8)$. So, eight architectures are required to get all the 8 outputs in parallel.

Since CLAs play a major role than CSAs in critical path, the objective of the proposed design is to reduce the number of CLAs in critical path and which will reduce the time delay. The proposed architecture follows signed carry save addition. Figure 4(a) shows the addition of 4, -6, 7, and -3 using signed carry save addition. Here 6^{th} bit of R will act as carry. Since four 3-bit values (Ex: 4, -6, 7, and -3) are added, the maximum possible output is 28 (7+7+7+7=28 which is 11100 in binary). Therefore, maximum possible bits of resultant R is 5. So 6^{th} bit of R will be the carry. Figure 4(b) shows the addition of -4, -6, 7, and -3 using signed carry save addition. In signed CSA, negative number can be accessed by taking 2's complement. During each addition, sign extension should be considered. The carry save adder needs 3 inputs to produce 2 outputs (sum and $carry$). Therefore, M input values can be added with $O(log_2 M)$ carry save stages. The final addition can be done by using recursive doubling based carry look ahead adder (CLA). The final carry ($Carry$) is used to find the resultant number ($Result$) is positive or negative. Figure 4(c) shows the carry save adder tree structure for 8 inputs ($p0$, $p1$, ...$p7$), which follows the Wallace tree multiplier [9] architecture.

III. The Proposed Architecture of Discrete Hadamard Transform

The proposed 4-point 1D-DHT architecture is shown in Figure 5, where 4 inputs are $x0$, $x1$, $x2$, and $x3$. Their corresponding sign bits are $x0s$, $x1s$, $x2s$, and $x3s$ respectively. In the first stage, inverters and CLAs are used to take 2's complement of input values. In the second stage, 2-to-1 multiplexers are used to select appropriate value from input values or 2's complemented input values. This selection will be based on the sign bits of input values. In the third stage, the outputs from appropriate 2-to-1 multiplexers are added with signed carry save reduction tree. This selection of output from particular 2-to-1 multiplexer is based on discrete Hadamard co-efficient matrix. For example, 4-point discrete Hadamard co-efficients for first row are 1, 1, 1, 1 and which is shown in equation (3). Therefore $z0$, $z1$, $z2$, and $z3$ are sent to signed carry save reduction tree 1.

Similarly 4-point discrete Hadamard co-efficients for second row are 1, -1, 1 and -1. Therefore $z0$, $z1r$, $z2$, and $z3r$ are sent to signed carry save reduction tree 2. The signed carry save addition is explained in Figure 4. In the last stage, CLA is used to add sum and $carry$ from carry save reduction tree. In the next step, $carry1$ is used to select the proper output (positive or negative). If $carry1 = 1$, then output is negative number otherwise output is positive. The $carry1$ will act as sign of resultant values $y0$, $y1$, $y2$, and $y3$. Table I shows number of n-bit CLA/CSA stages in worst path of different N-point 1D-DHT architectures, where inputs are considered as n-bits wide. The table clearly shows the proposed architectures requires less number of CLAs in critical path compared to other existing techniques.

Figure 7 shows proposed 4×4-point 2D-discrete Hadamard transform architecture, where the proposed 1D-DHT architecture is used. In general, 2D orthogonal discrete transform can be found in 2 steps, they are row process and column process. Here the input/output signal sample values and Hadamard co-efficients are represented as $N \times N$ matrices. During the row process, each row of input signal matrix is 1D transformed and the results are stored in $N \times N$ buffer. After completing all the N rows of input signal matrix, transpose matrix of the buffer is taken for column

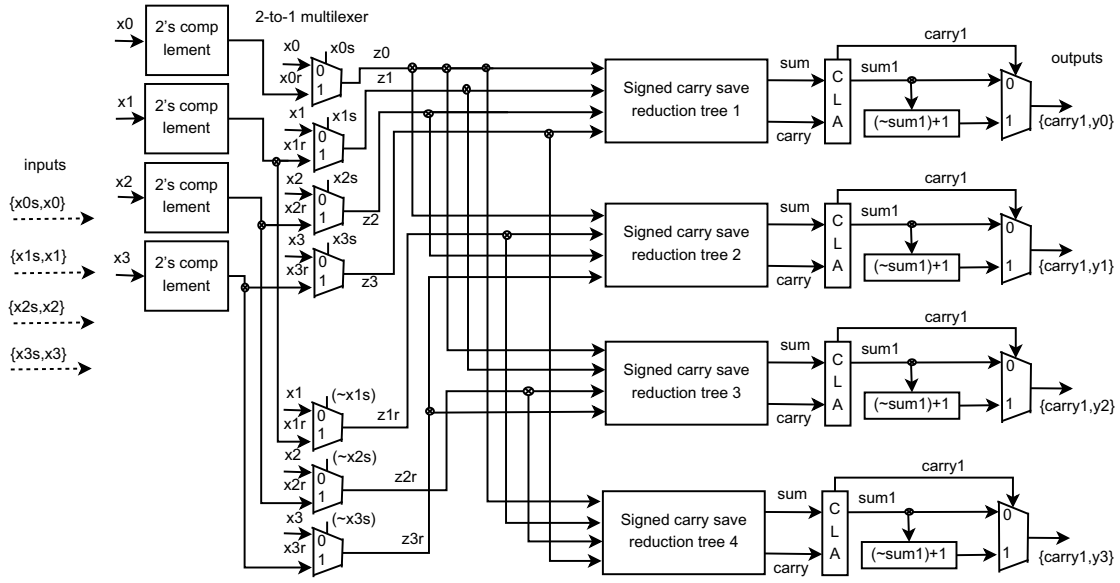

Figure 5. Proposed 4-point 1D-discrete Hadamard transform architecture

Table I
NUMBER OF n-BITS CLA/CSA STAGES IN WORST PATH OF DIFFERENT N-POINT 1D-DHT ARCHITECTURES WITH INPUTS OF n-BITS WIDE

	No. of CLA stages	No. of CSA stages	Others
Conventional [5]	$2(log_2 N)$	-	-
Linear systolic array [6]	$2(log_2 N)-2$	2	-
Distributed arithmetic [8]	4	-	T(LUT)
Proposed	2	$log_2 N$	-

process. In column process, each row of transposed buffer matrix is 1D transformed and results are the required 2D-transformed values. Figure 6 shows the 4×4-point 2D-DHT. During row process, each row of 4×4-input matrix is 1D transformed and the results are stored in each row of 4×4-buffer. During column process, each column of 4×4-buffer matrix is 1D transformed and the results are the required 2D-DHT values.

In general, two 1D-DHT's are used for row and column processes. Here one of the 1D-DHT will be idle during each process (row or column). So hardware utilization for each 1D-DHT is 50% during the entire process. This problem can be solved by using only one 1D-DHT hardware for both row and column processes. In this case, hardware utilization of 1D-DHT will be always full. In Figure 7, sel is used to select row or column process. During row process, inputs are taken from input signal matrix (input ports), which are represented as $x(i,0)$, $x(i,1)$, $x(i,2)$, and $x(i,3)$. Here i is varied from 0 to 3. During column process, inputs are taken form transposed 4×4 buffer matrix (feedback ports), which are represented as $z(0,i)$, $z(1,i)$, $z(2,i)$, and $z(3,i)$. The outputs of 4×4-proposed 1D-DHT are $y(i,0)$, $y(i,1)$, $y(i,2)$, and $y(i,3)$. During first clock cycle, $y(0,0)$ will be sent to the first row of transpose buffer. During second, third, and fourth clock cycles, $y(0,1)$, $y(0,1)$, and $y(0,3)$ will be

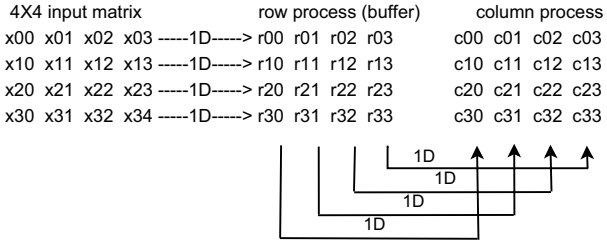

Figure 6. Example for row and column process of 4×4-point 2D-DHT

sent to the second, third, and fourth rows of transpose buffer respectively. Therefore a column of 4-to-1 multiplexers are used to receive the outputs from 1D-DHT. The 2-bit select line for all the 4-to-1 multiplexers will be incremented during each clock cycle by 2-bit up counter. The first four cycles are used to complete the four 1D-DHT's (1D-DHTs of four rows) of input matrix. Therefore, 8 clock cycles are required to complete 4×4-2D DHT. In general, $2N$ clock cycles are required to complete $N \times N$-2D DHT, where first N clock cycles are used for row process and second N clock cycles are used for column process. So the same proposed 1D-DHT hardware can be used to find 2D-DHT and the hardware utilization for proposed 1D-DHT is 100% during

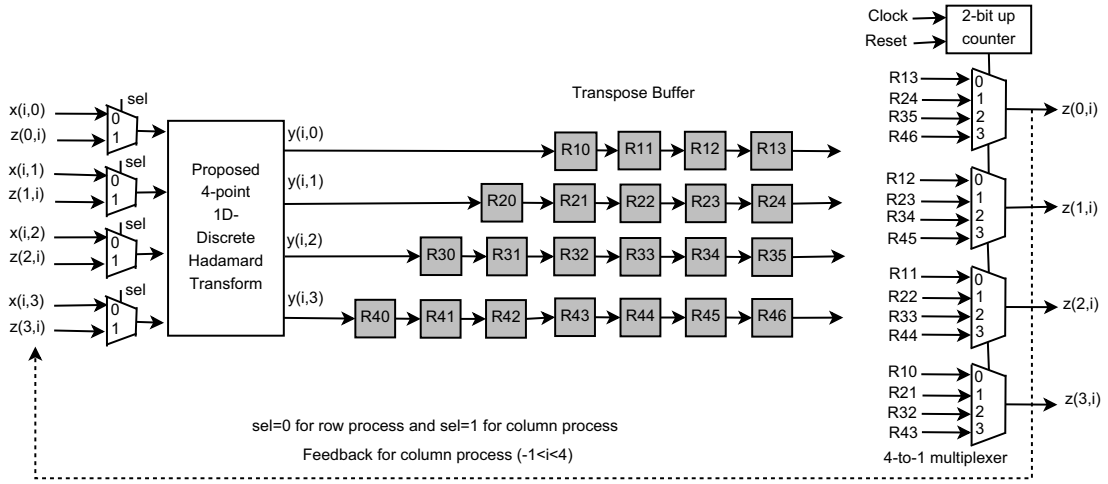

Figure 7. Proposed 4×4-point 2D-discrete Hadamard transform architecture

row and column processes. An another important thing with proposed 2D-DHT is that, the feasibility of 1D and 2D, which means that the proposed 2D-DHT architecture can act as 1D or 2D by making $sel = 0$ or 1 respectively. In Figure 7, hardware registers are represented as shaded boxes.

IV. Design Modeling, Implementation, and Results

All the existing and proposed designs are modeled in Verilog HDL. These Verilog HDL models are simulated and verified using Xilinx ISE simulator. The timing, area, and power analysis of this implementation are done with Cadence 6.1 ASIC design tool. All the designs are implemented for CMOS TSMC $45\ nm$ technology library, where $tcbn45gsbwpbc088\_ccs.lib$ is used as library. Here the operating voltage is $0.88\ v$. Table III shows the comparison of worst path delay, total area, net power, and power delay product (PDP) or energy per operation [12] between various 8-point 1D and 8×8-point 2D discrete Hadamard transform architectures. In this implementation, all the input signal sample values are considered as 32-bits wide.

Table II
Number of clock cycles for different architectures for 1D/2D DHT with inputs of n-bits wide

	N-point 1D DHT	$N \times N$-point 2D DHT
Conventional [5]	1	$2N$
Linear systolic array [6]	1	$2N$
Distributed arithmetic [8]	n	$2Nn$
Proposed	1	$2N$

The power delay product stands for the average energy consumed per switching event and it is apparent from the units ($W.s = Joule$). The PDP can be easily calculated by multiplying worst path delay with sum of switching and leakage power. Since the proposed architecture is designed for delay optimization, 55.14%, 42.07%, and 17.77% of improvement in worst path delay can be achieved by proposed 8-point 1D-DHT over conventional, linear systolic array, and distributed arithmetic based 1D-DHT architectures respectively. Here, distributed arithmetic based 1D-DHT architecture requires less delay then other existing techniques but it needs n-clock cycles to produce the output, where n-is the number of bits for input signal sample values. The number of clock cycles required for conventional, linear systolic array, and proposed 1D-DHT architectures is 1. Similarly proposed 8×8-point 2D-DHT achieves an improvement of 50.7%, 21.7%, and 51.17% in worst path delay over conventional, linear systolic array, and distributed arithmetic based 2D-DHT architectures respectively. In case of distributed arithmetic based 2D-discrete Hadamard transform [8], the feedback includes 1D-DHT (which includes accumulator) architecture along with counter for column process and this causes the worst path delay of distributed arithmetic based 2D-DHT to be higher than others. Here the number of clock cycles required for conventional, linear systolic array, and proposed $N \times N$-point 2D-DHT architectures is $2N$. In case of distributed arithmetic based $N \times N$-point 2D-DHT architecture, the number of clock cycles is $2nN$. Table II shows the number of clock cycles for different architectures for 1D/2D DHT. Figure 8 shows the layout diagram for proposed 8×8-point 2D-DHT architecture using 45-nm technology.

V. Conclusion

In this paper, a high performance VLSI architecture for N-point discrete Hadamard transform is proposed, where the proposed architecture is designed with signed carry save adder tree. So the depth of the architecture falls within the bounds of $O(log_2\ N)$. The same proposed architecture is implemented for $N \times N$-point 2D-discrete Hadamard trans-

Table III
PERFORMANCE ANALYSIS OF DIFFERENT ARCHITECTURES FOR DHT

1D/2D	DHT architecture	Worst path delay (ps)	Frequency (MHz)	Total area (μm^2)	Total no. of cells	Net power (nw)	Switching power (nw)	Leakage power (nw)	Energy per operation (fJ)
8-point 1D	Conventional [5]	1778.7	562.20	23359	18732	1063478.98	3624992.59	1279402.25	8723.45
8-point 1D	Linear systolic array [6]	1378.2	725.58	19486	15178	906461.72	3157993.38	1089286.41	5853.6
8-point 1D	Distributed arithmetic [8]	959.5	1042.2	21827	16688	146608.74	496817.94	1077166.9	1510.25
8-point 1D	**Proposed**	**798**	**1253.1**	**16606**	**11519**	**925368.27**	**3352081.11**	**1019842.23**	**3488.79**
8 X 8-point 2D	Conventional [5]	2366.4	426.6	41592.4	25772	890730.2	3047381.5	2019256.9	11989.6
8 X 8-point 2D	Linear systolic array [6]	1678.4	595.8	35392.7	19834	702322.2	2575086.6	1722929.2	7213.7
8 X 8-point 2D	Distributed arithmetic [8]	2382.8	419.7	36164.3	20777	423275.6	1578799.8	1649671.1	7692.8
8 X 8-point 2D	**Proposed**	**1165.2**	**858.2**	**32317.9**	**16023**	**688887.9**	**2622005.9**	**1615466.9**	**4937.5**

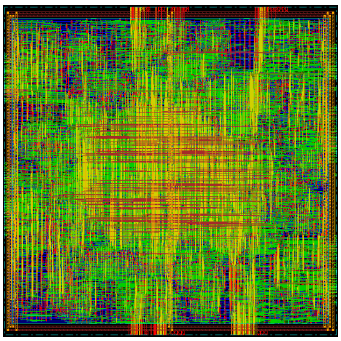

Figure 8. Layout chip diagram for proposed 8×8-point 2D-DHT with core area as $35549.7\mu m^2$, die space around core as $5\mu m$, and total chip area as $39420.7\mu m^2$ using $45\ nm$ technology.

form, where row/column processing is carried with same 1D-DHT hardware. The proposed 8-point 1D-DHT achieves an improvement of 55.14% and 42.07% in worst path delay over conventional and linear systolic array based 1D-DHT architectures respectively. Similarly, the proposed 8×8-point 2D-DHT achieves an improvement of 50.7%, 21.7%, and 51.1% in worst path delay over conventional, linear systolic array, and distributed arithmetic based 2D-DHT architectures using $45\ nm$ CMOS TSMC library respectively.

ACKNOWLEDGMENT

This work is funded under the SMDP-C2SD project by Department of Electronics and Information Technology (DeitY), Govt. of India.

REFERENCES

[1] Steven W. Smith, "*The Scientist and Engineers Guide to Digital Signal Processing*", California Technical Publishing, 1997, pp. 551-566.

[2] Jin-il Flyun, Jin-Jong Cha, In Kang, Jaeseok Kim, and Kyungsoo Kim , "*Reverse Link Demodulator ASIC for CDMA Cellular System*", IEEE International Symposium on Circuits and Systems, vol. 4, pp. 276-279, May 1996.

[3] Ihab Amer, Wael Badawy, and Graham Jullien , "*A VLSI Prototype for Hadamard Transform with Application to MPEG-4 Part 10* ", IEEE International Conference on Multimedia and Expo, vol. 3, pp. 1523-1526, June 2004.

[4] Pratt, W. Kane, J. and Andrews, "*Hadamard transform image coding*", Proceedings of the IEEE, vol. 57, pp. 58-68, June 2005.

[5] Jialiang Liu, Xinhua Chen, Yibo Fan and Xiaoyang Zeng, "*A Full-mode FME VLSI Architecture Based on 8x8/4x4 Adaptive Hadamard Transform For QFHD H.264/ A VC Encoder*", IEEE International Conference on VLSI and System-on-Chip, pp. 434-439, Oct. 2011.

[6] P. K. Meher and J. C. Patra, "*Fully-Pipelined Efficient Architectures for FPGA Realization of Discrete Hadamard Transform*", IEEE International Conference on Application-Specific Systems, Architectures and Processors, pp. 43-48, July 2008.

[7] A. Amira, A. Bouridane and P. Milligan, "*Novel Architecture for Walsh Hadamard Transforms Using Distributed Arithmetic Principles*", IEEE International Conference on Electronics, Circuits and Systems, vol. 1, pp. 182-185, Dec. 2000.

[8] A. Amira and S. Chandrasekaran, "*Power Modeling and Efficient FPGA Implementation of FHT for Signal Processing*", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 15, no. 3, pp. 286-295, Mar. 2007.

[9] C. S. Wallace, "*A suggestion for a fast multiplier*", IEEE Transactions on Electronic Computers, vol. EC-13, no. 1, pp. 14-17, Feb. 1964.

[10] Tu-Chih Wung, Yu- Wen Huang. Hung-Chi Fang, and Liang-Gee Chen , "*Parallel 4X4 2D transform and inverse transform architecture for MPEG-4 AVC/H. 264*", IEEE International Symposium on Circuits and Systems, vol. 2, pp. II-800-II-803, May 2003.

[11] Basant Kumar Mohanty, Pramod Kumar Meher, and Subodh Kumar Singhal, "*Efficient Architectures for VLSI Implementation of 2-D Discrete Hadamard Transform*", IEEE International Symposium on Circuits and Systems, pp. 1480-1483, May 2012.

[12] Ricardo Gonzalez, Benjamin M. Gordon, and Mark A. Horowitz, "*Supply and Threshold Voltage Scaling for Low Power CMOS*", IEEE Journal of Solid State Circuits, vol. 32, no. 8, pp. 1210-1216, Aug. 1997.