International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Efficient Hybrid VLSI Architecture Using Hadamard Transform

Nitesh Dodkey
TIT, Bhopal

Prof. Divya Jain
TIT, Bhopal

Prof. Vikas Gupta
TIT, Bhopal

Abstract **– Fully-pipelined and parallel modular structures are presented in this paper for efficient hardware realization of discrete Hadamard transform (HT). From the kernel matrix of HT, we have derived four different pipelined modular designs for transform length N = 4. It is shown further that the HT of transform-length N = 8 can be obtained from two 4-point HT modules, and similarly, the HT of transform-length N=16 can be obtained from four 4-point HT modules. Long-length transforms may, however, be computed from these short-length modules as N-point transforms can be computed from 2M number of M point HT-modules, where $M = N^{1/2}$. The proposed architectures are coded in VHDL, simulated by Xilinx ISE tool for validation and testing; and synthesized thereafter to be implemented in FPGA device Virtex-E. From the synthesis result, it is found that the pro-posed designs involve considerably less number of slices.**

Keywords **– Hadamard Transform, Parallel Processing, Pipelining, FPGA, DOT.**

I. Introduction

The Hadamard transform (HT) is one of the frequently encountered discrete orthogonal transforms (DOT) [1–6]. It belongs to the class of rectangular real-valued DOT, which can be used for various basic digital signal processing (DSP) operations including the computation of the discrete Fourier transform (DFT), implementation of digital filters and spectral estimation [2]. It is considered as an important tool for speech processing and error control coding [3, 4]. Not only is it used frequently for audio and video processing, but also has a lot of other applications in signal and image processing [7,8]. The 2-dimensional (2-D) HT based on its 1-D counterpart is well known for its application in lossless image compression [3]. The simplicity and orthogonality of the HT has been utilized for efficient generation of pseudo noise sequences for code division multiple access (CDMA) technique in spread spectrum communication systems [9, 10]. Recently, it is also found to have application in quantum information processing and quantum cryptography, where HT is used as a quantum gate [5]. Several other applications of HT and its variations may be found in [2].

Apart from its wide application domain, HT is popular for the simplicity and regularity of its kernel matrix con-sisting only of 1s and -1s. The simplicity of kernel of HT leads to reduced arithmetic-complexity and efficiency of hardware and software implementations compared with other sinusoidal DOTs like DFT, discrete cosine transform (DCT) and discrete sine transform (DST). Unlike the sinusoidal DOTs, it does not involve any multiplication operations, and can be computed by additions and subtractions only. In order to reduce the number of additions, several fast algorithms have been proposed in [2] and [11]. Along with the growing commercial interest in integrating various signal and image processing systems for high-volume applications, there has been a considerable interest to design dedicated circuits for hardware implementation of HT. Field programmable gate array (FPGA) devices are becoming increasingly popular in the last a few years, not only due to their lower cost compared with the application specific integrated circuits (ASIC) but also for their scope for reusability as programmable device and increasing transistor density available in it to implement various complex algorithms in signal processing, image processing and communication. In recent years, we therefore observe a growing interest for efficient implementation of various computation-intensive algorithms including the HT in FPGA platform [10, 12,]. In this paper, we propose simple and efficient pipelined de-signs for implementation of HT for transform length N=4, 8 and 16, which can be used to generate HT of higher transform-lengths.

The rest of the paper is organized as follows. The algorithms of FPGA implementation of HT are derived in Section 2, and the proposed architectures are described in Section 3. The implementation results are given in Section 4 and conclusions are outlined in Section 5.

II. Mathematical Formulation

The HT of a sequence {x(n) for n = 0, 1, 2, ..., N − 1} may be defined as

$$X(k) = \sum_{n=0}^{N-1} (H_N(k, n).x(n) \quad (1)$$

for $0 \le k \le N - 1$, where H_N is a Hadamard matrix of size N × N, which is defined recursively as

$$H_N = H_{N/2} \ □ \ H_2 \quad (2)$$

where □ represent the kronecker product

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (3)$$

Hadamard matrix $\mathbf{H}_N$ may therefore be given by

$$H_N = \begin{pmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{pmatrix} \quad (4)$$

Using (4), the HT of N-point sequence {x(n)}, given by (1) can be written as,

$$\begin{pmatrix} X_2^1 \\ X_2^2 \end{pmatrix} = H_{N/2} \otimes \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} X_4^1 \\ X_4^2 \\ X_4^3 \\ X_4^4 \end{pmatrix} = H_{4/2} \otimes \begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad (9)$$

Where

X_4^1=[X(1) X(2)………….$X(N/4-1)]^T$
X_4^2=[X(N/4) X(N/4)……..$X(N/2-1)]^T$
X_4^3=[X(N/2) X(N/2+1)….$X(3N/4-1)]^T$
X_4^4=[X(3N/4) X(3N/4+1)...$X(N-1)]^T$ (10)

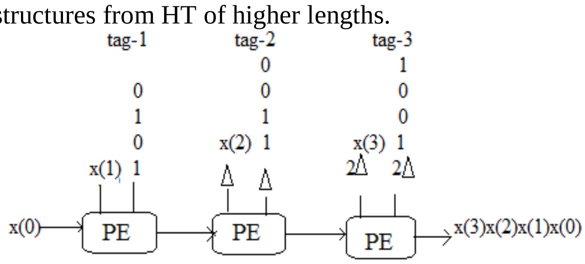
Such that

$X_4^1=H_{N/4}(x_4^1+x_4^2+x_4^3+x_4^4)$ (11a)
$X_4^1=H_{N/4}(x_4^1-x_4^2+x_4^3-x_4^4)$ (11b)
$X_4^1=H_{N/4}(x_4^1+x_4^2-x_4^3-x_4^4)$ (11c)
$X_4^1=H_{N/4}(x_4^1-x_4^2-x_4^3+x_4^4)$ (11d)

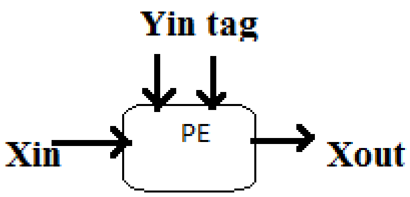
It may be noted from (7) that the HT of an *N*-point sequence could be obtained from two *N*/2-point sequences derived from the original sequence. Similarly, from (11), we find that *N*-point HT can also be obtained from four *N*/4-pint HT of four sequences derived from the original sequence by suitable add/subtract operations. The recursive behaviour of Hadamard matrix can therefore be utilized to compute an *N*-point HT by 2-point or 4-point HT. In the next Section we have shown that 4-point HT is fairly simple to implement in a single PE to be used further for the computation of HT of higher lengths.

III. Proposed Pipelined and parallel Architecture for FPGA Implementation

Some different computing structures are derived here for 4-point HT, and used them further to generate the efficient structures from HT of higher lengths.



(a) Proposed 4 point linear HT structure



If tag=1 $X_{out}=X_{in}+Y_{in}$
Else $X_{out}=X_{in}-Y_{in}$

(b) Function of the PEs.

Fig-1

3.1 Structure for 4-point HT

The HT of a 4-point sequence {x(0), x(1), x(2), x(3)} can be obtained from (1) and (2) as

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Since the HT kernel given by (12) involves only 1s and -1s, it would be possible to design simple adder-based modules for computing the 4-point HT shown in Figs.1-3.

The single-array structure of Fig.1 consists of 3 PEs, where each PE either performs an addition (for tag=1) or a subtraction (for tag=0). The function of each PE is described in Fig.1(b).The tag-bits may be fed to the individual PEs through 4-bit shift-registers. But, it is more efficient to use a tag-bit generator to obtain the necessary tag-bits for the PEs. Each of the PEs is fed with the input sample in a staggered manner to maintain the necessary data-dependence requirement. It will yield the first output after three cycles, and it will produce a throughput of 1 output in every cycle, thereafter. The duration of cycle period of the PEs is $T = T_A$, where T_A is the time required for an addition operation.

To have higher throughput a 2-dimensional (2-D) structure for the computation of 4-point HT [shown in Fig.2] can be used instead of linear array design of Fig.1. The 2-D structure of Fig.3 consists of eight 2point 1D hadamard structure. The proposed 2-D structure yields its first out-put three cycles after the first input arrives at the structure and yields four output during every cycle after the horizontal and vertical pipelines are filled in the first seven cycles. The duration of cycle period of the 2-D structure is $T = T_A$ (the same as that of linear array of Fig.1). The linear array of Fig.1 can be extended further for any higher-length HT, so that a linear array consisting of $(N-1)$ PEs can be used for *N*-point HT. Similarly, the 2-D array of Fig.2 can also be extended further for higher-length HT, so that a 2-D array consisting of $N(N-1)$ add/subtract cells can be derived accordingly for *N*-point HT.

Using the kernel values of (12), we can also have a two-stage algorithm for computation of 4-point HT as follows.

3.1.1 Step 1:

Compute the following in parallel:

$$\begin{aligned} a(0) &= x(0) + x(1) \\ a(1) &= x(2) + x(3) \\ a(2) &= x(0) - x(1) \\ a(3) &= x(2) - x(3) \end{aligned} \qquad (13)$$

3.1.2 Step 2:

Compute the following in parallel:

$$\begin{aligned} X(0) &= a(0) + a(1) \\ X(1) &= a(0) - a(1) \\ X(2) &= a(2) + a(3) \\ X(3) &= a(2) - a(3) \end{aligned} \qquad (14)$$

Based on the two-stage algorithm of eq.(13) and (14) two pipelined structures [shown in Figs.3] are derived for four-point HT. The transverse pipelined structure for 4-point HT is shown in Fig.3. The transverse pipelined structure of Fig.3 consists of an addition module (AM), subtraction module (SM) and a pair of add-subtract module (AS). The AM consists of two adders, while SM consists of two subtractors. Each AS module consists of an adder and a subtractor. During every cycle the 4-point.

3.4.2 Structure for 2-point 2-D hadamard transform

Using the procedure explained above

Input matrix: $[X]=\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}$

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Result after row wise transformation

$$[U] = \begin{pmatrix} x_{11} + x_{12} & x_{21} + x_{22} \\ x_{11} - x_{12} & x_{21} - x_{22} \end{pmatrix}$$

Result after column wise transformation

$$[V] = \begin{pmatrix} x_{11} + x_{12} & x_{21} + x_{22} \\ x_{11} - x_{12} & x_{21} - x_{22} \end{pmatrix}$$

It can be seen that u11 and u21 are the 2-point 1-D hadamard transform results of inputs x11 and x12. Similarly u12 and u22 are the the 2-point 1-D hadamard transform of x21 and x22.

Result after column- wise calculation:

$$[V] = \begin{pmatrix} (x_{11} + x_{12}) + (x_{21} + x_{22}) & (x_{11} - x_{12}) + (x_{21} + x_{22}) \\ (x_{11} + x_{12}) - (x_{21} + x_{22}) & (x_{11} - x_{12}) - (x_{21} + x_{22}) \end{pmatrix}$$

= [Y] = {output matrix}

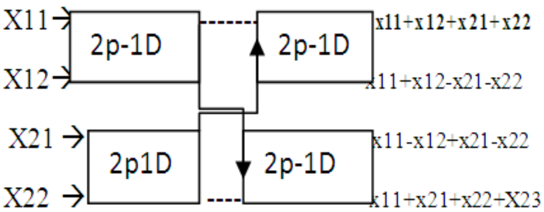


Fig.2.

On the basis of above 2point 2D hadamard structure of 4 point, 8 point structure can be made. It can be seen that Long-length transforms can be computed from these short-length modules as N-point transforms can be computed from 2M number of M point HT-modules, where $M = N^{1/2}$.

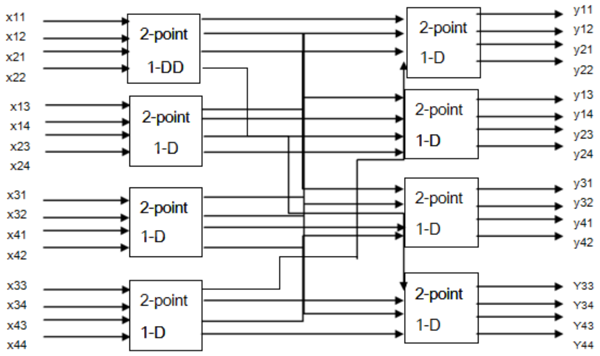


Fig.3. Structure of 4 point 2D architecture

IV. Simulation Results

To study the performance of the proposed architectures of HT, the proposed design is coded in VHDL and simulated by using Xilinx ISE tool. The word-length of both input and the output words are taken to be 4-bit wide. The power consumption of the proposed design are obtained from the synthesis results for N. The Power of the structures in terms of best-achievable-frequency (BAF) is obtained after performing place-and-route by using the place-and-route tool available in the Xilinx ISE package. The timing results of pipelined and proposed parallel structure are listed in Table.

Device	Length	Proposed	Structure Of 1
		Slices	Slices
Virtex-E	4	22	32
	16	177	256
	64	710	-

V. Conclusion

Simple and efficient fully-pipelined and parallel structures are de-signed for implementation of the HT. Four different pipelined and parallel modules designs are presented for transform length *N* = 4. It is shown that the HT of transform-length *N* = 8 can be obtained from two 4-point HT modules and the HT of transform-length *N* = 16 can be obtained from four 4-point HT modules. Higher length transforms may however be computed from these short-length modules as *N*-point transforms can be computed from 2*M* number of *M* point HT-modules, where $M = N^{1/2}$. The proposed architectures are coded in VHDL, simulated by Xilinx ISE tool for validation, and synthesized for virtex-E device available Xilinx tool.

References

[1] Fully-Pipelined Efficient Architectures for FPGA Realization of Discrete Hadamard Transform

[2] K. G. Beauchamp. Applications of Walsh and Related Func-tions. Academic, New York, 1984.

[3] F. J. MacWilliams and N. J. A. Sloane. The Theory of Error-Correcting Codes. North-Holland, T. Amsterdam, The Netherlands, 1992.

[4] M. Lukac and M. Perkowski. Evolving quantum circuits using genetic algorithm. In Proceedings NASA/DoD Conf. Evolvable Hardware, pages 177–185, May 2002.

[5] L. W. Chang and M. C. Wu. A bit level systolic array for Walsh-Hadamard transforms. Signal Process., 31(3):341– 347, Apr. 1993.

[6] A. Amira, A. Bouridane, P. Milligan, and M. Roula. NTSC component separation via Hadamard transform. IEE Proceedings-Vision, Image and Signal Processing,, 141(1):27–32, Feb. 1994.

[7] Y.-W. Huang, B.-Y. Hsieh, T.-C. Chen, and L.-G. Chen. Hardware architecture design for H.264/AVC intra frame coder. In Proceedings 2003 International Symposium on Circuits and Systems, (ISCAS '03), volume 2, pages 269– 272, May 2004.

[8] J. Flyun, J.-J. Cha, I. Kang, J. Kim, and K. Kim. Reverse link demodulator ASIC for CDMA cellular system. In Proc. 1996 IEEE International Symposium on Circuits and Systems, (ISCAS '96), 'Connecting the World',, volume 4, pages 276–279, May 1996.

[9] S. K. Bahl. Design and prototyping a fast Hadamard trans-former for WCDMA. In Proceedings 14th IEEE Int. Work-shop on Rapid Systems Prototyping, pages 134–140, June 2003.

[10] A. Elnaggar and M. Aboelaze. A new recursive formulation for 2-D WHT. In Proceedings 2003 International Symposium on Circuits and Systems, (ISCAS '03), volume 4, pages 484–487, May 2004.

[11] A. Amira, A. Bouridane, P. Milligan, and M. Roula. Novel FPGA implementations of walsh-hadamard transforms for signal processing. IEE Proceedings-Vision, Image and Sig-nal Processing, 148(6):377–383, Mar. 2001.

[12] A. Amira and S. Chandrasekaran. Power modelling and efficient FPGA implementation of FHT for signal processing. IEEE Trans. Very Large Scale Integration (VLSI) Systems, 15(3):286–295, Mar. 2007.

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

AUTHOR'S PROFILE

Nitesh Dodkey

presently perusing ME in Digital Communication from RGPV university. He has completed Bachelor of Engineering in Electronics and Communication stream. His research interest include custom computing using field-programmable gate arrays (FPGAs), signal processing, low-power architectures for communications and digital circuits and systems.

Prof. Divya Jain

presently Perusing Ph. D. from RGPV University. She has completed M.Tech with specialisation in Digital Communication and bachelor in Engineering with distinction .Her Research interest include digital image processing ,signal processing, OFDM based wireless LAN. She has published number of papers in many national and international publications. She has attended lots of seminars and workshop in various universities.

Prof. Vikas Gupta

HOD of EC Department in Technocrats Institute of Technology, Bhopal and presently perusing Ph.D from MANIT, he has completed M.Tech in Digital Communication from MANIT Bhopal. His research interest include digital signal processing, image processing .He has published numbers of national and international papers in various journals.