

Hardware design of a scalable and fast 2-D hadamard transform for HEVC video encoder

Heh Whit Ney¹, Ab Al-Hadi Ab Rahman², Ainy Haziyah Awab³, Mohd Shahrizal Rusli⁴,
Usman Ullah Sheikh⁵, Goh Kam Meng⁶

^{1,2,3,4,5}School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia

⁶Faculty of Engineering and Technology, Tunku Abdul Rahman University College, Malaysia

Article Info

Article history:

Received Nov 30, 2018

Revised Mar 25, 2019

Accepted Apr 3, 2019

Keywords:

FPGA

Hadamard transform

HEVC

RTL

SATD

ABSTRACT

This paper presents the hardware design of a 2-dimensional Hadamard transform used in the rate distortion optimization module in state-of-the-art HEVC video encoder. The transform is mainly used to quickly determine optimum block size for encoding part of a video frame. The proposed design is both scalable and fast by 1) implementing a unified architecture for sizes 4x4 to 32x32, and 2) pipelining and feed through control that allows high performance for all block sizes. The design starts with high-level algorithmic loop unrolling optimization to determine suitable level of parallelism. Based on this, a suitable hardware architecture is devised using transpose memory buffer as pipeline memory for maximum performance. The design is synthesized and implemented on Xilinx Kintex Ultrascale FPGA. Results indicate variable performance obtained for different block sizes and higher operating frequency compared to a similar work in literature. The proposed design can be used as a hardware accelerator to speed up the rate distortion optimization operation in HEVC video encoders.

*Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Ab Al-Hadi Ab Rahman,
School of Electrical Engineering, Faculty of Engineering,
Universiti Teknologi Malaysia,
81310 Skudai, Johor Bahru, Malaysia, +6019-7338914.
Email: hadi@fke.utm.my

1. INTRODUCTION

In accordance with the advancement of multimedia technology, the demand for higher video resolution is growing. High definition video has become a basic expectation among consumers and they continue to push for better and smoother viewing experience. In relation, video coding standard has evolved greatly from the early MPEG standard to the H26x family. High Efficiency Video Coding (HEVC) or H.265 is the latest standard from the H26x family. Being the state-of-the-art video coding standard, it offers an identical quality to the previous Advanced Video Coding (AVC) or H.264 standard, but only requires half the bitrate of AVC [1-3].

The improvement in coding efficiency is contributed mainly by the advancement in the video compression method, i.e. intra-frame prediction. Intra-frame prediction compresses a frame by looking for redundant information in the same frame. While AVC only supports 9 prediction modes, HEVC is able to support a total of 35 prediction modes. With more prediction angles, the prediction is much more accurate, less redundant and hence massively reduce the size of bits required to encode each frame [4]. Lesser bits means smaller file size and reduced bandwidth requirement. Every prediction unit has to go through all the prediction modes to determine which is the best suited prediction mode. These modes are evaluated by a cost function calculated by Sum of Absolute Transformed Difference (SATD), a mathematical method used in

fast Rate Distortion Optimization (RDO). The core operation of the SATD for RDO is the Hadamard Transform, which is the main focus of this paper.

Although the increase in the number of prediction modes enhance the compression quality, at the same time, they came at a price: substantial computational complexity. To overcome the limitation caused by the increased complexity, the Hadamard transform in the SATD must be accelerated in hardware with the capability of handling variable block size. From hardware perspective, the architectural design of the Hadamard transform and the control of the architecture determines the performance of the HEVC encoder [5]. Research on SATD is mainly reported in literature for HEVC intra prediction as given by [6, 7, 8, 9].

The following reviews some of the related works on hardware architectures of the Hadamard transform for SATD. The work in [10] proposes six architectures based on the Fast Hadamard Transform (FHT) butterfly and Transform Exempted (TE) SATD. The authors show that by using FHT, the computational complexity is reduced to $2n^2 \log(n)$ as compared to n^3 using naive matrix multiplication. An improvement to this is given in [11] when the TE SATD merges the absolute sum calculation into the second 1-D FHT, results in more energy efficient design. These works were synthesized to 45nm ASIC technology. Other works include algorithms development for SATD [12, 13] and efficient hardware architectures [14-16]. The work in [17] presents an approach to explore the design space of HEVC transform. Some of the design methods in these works were incorporated in the present paper.

The work in [18] presents an architecture for an 8x8 Hadamard transform, mainly used for ultra-high resolution video sequence. Compared to a 4x4 transform, the 8x8 transform results in 9% to 19% less total execution time for these types of video. In the implementation, the authors split the Hadamard transform into two stages of 1-D transform. Pipeline registers are used between stages for higher performance. The designs are synthesized and implemented on both ASIC 45nm and Altera Stratix III FPGA. This is also similar to the work in [19] that focuses on size 8x8 only. The work in [20] attempts at designing all sizes from 4x4 to 32x32. However, the units are designed separately with the main objective is to evaluate the growth impact of the transpose buffers and how linear buffers could be used instead for better resource utilization. This work was also synthesized to ASIC 45nm technology. Other related works in the design of Hadamard transform include [21] for AVC encoders, and [22] on new methods and algorithms.

The issue with transpose versus linear buffer was also studied in [23] where it concludes that transpose buffer consumes large amount of power with tradeoff for performance. Linear buffer on the other hand results in power efficiency but low throughput. The authors presented new methods to efficiently control the linear buffer using a dedicated finite state machine. Furthermore, the study also concluded that for small size transform, the performance of transpose and linear buffers are almost identical. As the transform size increase, the the transpose buffer shows exponential growth in performance.

The work in [24] implements a unified architecture for the Hadamard transform size from 2x2 to 8x8. The smallest 2x2 unit is being reused to implement 4x4 and 8x8. However, the authors did not implement size 16x16 and 32x32 which can provide a greater compression efficiency. Another issue with the design is the fixed latency for all input block size, where a size of 2x2 has to go through the 32x32 structure. This work however, was improved in [25] where a new adder-subtractor compressor was implemented to achieve better power efficiency.

The work in the present paper proposes two contributions that are not described in any of the previous works: 1) a unified architecture for variable block sizes from 4x4 to 32x32, and 2) throughput optimization by pipelining and feedthrough control. Pipelining results in higher performance, while feedthrough control allows variable throughput depending on the input block size. The designs are synthesized and implemented on Xilinx Kintex Ultrascale device, and compared with the work in [18] using Altera FPGA.

2. RESEARCH METHOD

This section presents basic theory on the Hadamard transform, its algorithm, and our proposed hardware architecture.

2.1. SATD and Hadamard Transform

SATD calculation for an $M \times N$ block is defined as follows:

$$c \times \sum_{i=1}^M \sum_{j=1}^N |t^d_{ij}| \quad (1)$$

and,

$$T^D_{n \times n} = T_{n \times n} \times D'_{n \times n} \times T^T_{n \times n} \tag{2}$$

Where $t^{d}_{ij} \in T^D$ is a 2-D transformed difference block, and c is a scaling constant. c is typically given the values 1, 0.5, 0.25, 0.125 and 0.0625 for transform sizes of 2x2, 4x4, 8x8, 16x16, and 32x32 respectively. In equation 2, T represents a transformation matrix of an integer linear transform, normally a Walsh-Hadamard Transform [12]. D' represents the differences matrix obtained between an original block O and a candidate block C . It is also commonly known as the residual block. The Hadamard matrix can be grown accordingly for order higher than four as in Figure 1. The Hadamard matrix has a recursive definition for size $2_n \times 2_n$ as given by:

$$H_{2n \times 2n} = H_{2 \times 2} \otimes H_{n \times n} \tag{3}$$

The symbol \otimes is the Kronecker product and $H_{2 \times 2}$ is the lowest-order Hadamard matrix given by:

$$H_{2 \times 2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4}$$

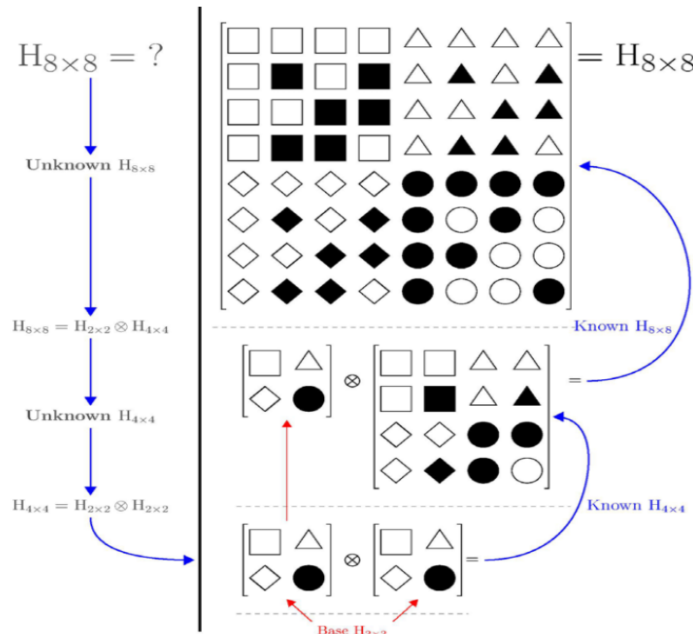


Figure 1. Method to grow the Hadamard matrix from size 2x2 to 32x32

2.2. SATD Algorithm and Loop Unrolling

The SATD algorithm is given in Figure 2, where $w[i]$ is the i^{th} element of a residual block and $x[i]$ represents the transformed value of $w[i]$. From equation 2 for a 2x2 matrix, it can be observed that the operation constitutes four operations which can be executed in parallel. Hence the algorithm is best to be unrolled by a factor of four. The new pseudo code with an unroll factor of four is shown in Figure 3. For calculating the transform difference of 4x4 matrix, a 2x2 transform must be performed prior to 4x4. Similarly 4x4 should be calculated first in order to obtain 8x8, until size 32x32.

2.3. Proposed Architectures

The proposed datapath architecture of the 4x4 2-D Hadamard transform module is given in Figure 4. It comprises of four main parts: first 1-D transform, shift registers, second 1-D transform and summation. These four basic components are applicable to sizes beyond 4x4 as well. For 4x4 transform, the incoming residual block contains 16 elements. The module can accept 4 elements in one clock cycle at input $w_{3:0}$. Hence, to process 16 elements, 4 clock cycles are required. A group of 4 elements will be stored into pipeline register $w_{3:0}$ in the next clock cycle. These values will go through the first round of transform in the smaller 2x2 block. After first transformation, the values are stored into shift register array denoted by x .

```

inputs: w[0:size-1];
outputs: satd_sum;
int satd = 0; int i = 0;
while (i < size) {
    x[i] = HT(w[i]);
    satd = satd + abs(x[i]);
    i = i + 1;
}
if (size != 2) {
    satd_sum = satd / (0.5 * sqrt(size));
} else {
    satd_sum = satd;
}
    
```

Figure 2. SATD algorithm with 2x2 Hadamard transform without loop unrolling

```

inputs: w[0:size-1];
outputs: satd_sum;
int satd = 0; int i = 0; int n = size/4;
while (i < n) {
    x[i] = (w[i] + w[i+n]) + (w[i+2*n] + w[i+3*n]);
    x[i+n] = (w[i] - w[i+n]) + (w[i+2*n] - w[i+3*n]);
    x[i+2*n] = (w[i] + w[i+n]) - (w[i+2*n] + w[i+3*n]);
    x[i+3*n] = (w[i] - w[i+n]) - (w[i+2*n] - w[i+3*n]);
    satd = satd + abs(x[i]) + abs(x[i+n])
            + abs(x[i+2*n]) + abs(x[i+3*n]);
    i = i + 1;
}
if (size != 2) {
    satd_sum = satd / (0.5 * sqrt(size));
} else {
    satd_sum = satd;
}
    
```

Figure 3. SATD algorithm with 2x2 Hadamard transform with loop unrolling

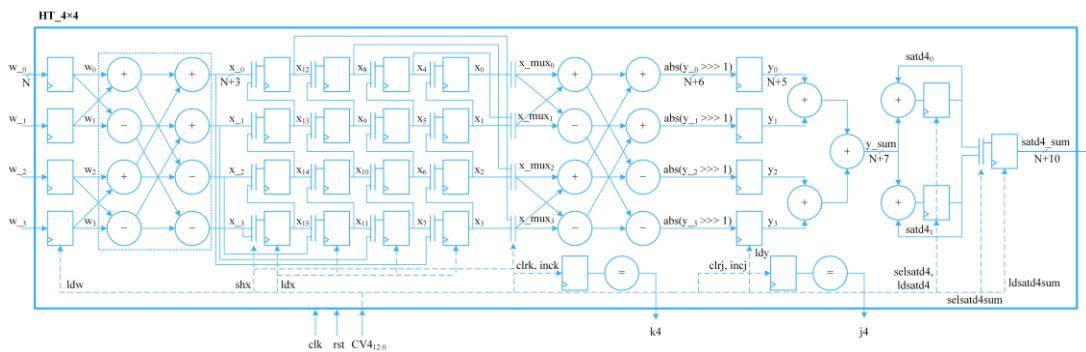


Figure 4. Proposed architecture for 4x4 2-D Hadamard transform

The shifting behaviour of x is illustrated in Figure 5. Shift register array will shift their data in either right or top direction based on the control signal shx . For fresh start, registers will shift their data to the right.

After four clock cycles, all the 16 inputs are transformed and stored in $x_{15:0}$. x_0, x_4, x_8 and x_{12} are required to compute the second transformation for the first group of data. The multiplexer x_{mux} will select these values, which are located in the first horizontal row of the shift register array, and feed them into the second transform. Note that the shift registers here essentially acts also as pipeline registers.

Similar to the 4x4 transform, the 8x8 transform also comprises of four main parts: first 1-D transform, shift registers, second 1-D transform and summation. Since 8x8 SATD can accept 8 inputs in one clock cycle, two 4x4 modules are required.

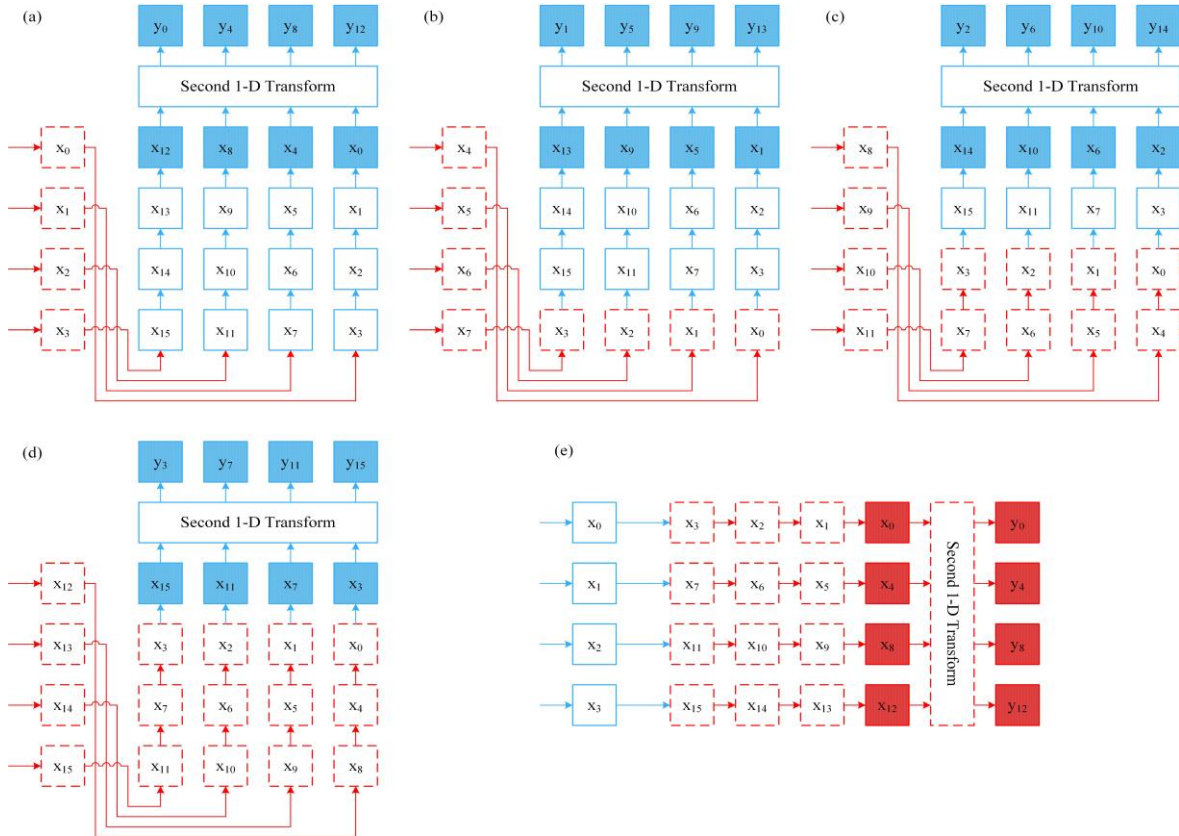


Figure 5. Shifting behavior of the proposed 4x4 2-D Hadamard transform architecture

Since a 8x8 2-D Hadamard transform module is built, the hardware must support both 4x4 and 8x8 operations. If the incoming block is a 8x8 residual block, the transformation will utilize all 35 components. However, if the incoming block is just a 4x4 residual block, $satd4_sum$ will be available after 2-D transformation is conducted in one of the 4x4 module. Hence, there is no need for the sum to go through further transformation. Instead, the sum can be copied directly to the final output register of the module, and in this case the register is $satd8_sum$. Such operation can be described as signal bypassing the data path and is known as feedthrough control. The register used to hold the feedthrough value is denoted as $satd4sumft$ as shown in Figure 6. Sizes beyond 8x8 have similar components to 8x8 transform. A simplified 32x32 transform datapath is shown in Figure 7.

3. RESULTS AND ANALYSIS

The proposed architectures in section 2.3. have been specified using Verilog HDL, synthesized and implemented using Xilinx Kintex Ultrascale FPGA device. The results are analyzed in terms of resource, timing, throughput and power.

The resources used by 4x4, 8x8, 16x16 and 32x32 are reported in Table 1. For Look-up Table (LUT), the usage has increased by three to four fold when the size of the transform increases. Usage of Flip-Flop (FF) is increasing at factor of 5 to 6 times as well. This is expected since the number of shift registers

increases from 16 to 1024. Number of IO has grown from 34 to 186 and this is also expected. The number of inputs must scale accordingly to the increased transform size.

Table 1. Resource Utilization of the Hardware Hadamard Transform

Size	LUT	FF	IO	BUFG
4x4	329	216	34	1
8x8	1187	1191	58	1
16x16	4686	6146	102	3
32x32	19736	30157	186	5

Timing is an important metric that would determine the performance of a design. This design contains a single clock denoted as *clk* that synchronizes the data flow of the hardware. As shown in Figure 8, the maximum operating frequency is reduced from 321.65 MHz for 4x4 to 230.95 MHz for 32x32. These values are obtained by taking the reciprocal of the data arrival time of each design. An interesting observation from this graph is the small change in frequency between 16x16 and 32x32 designs. This is due to the saturated critical path for both of the designs, hence giving an almost identical maximum frequency.

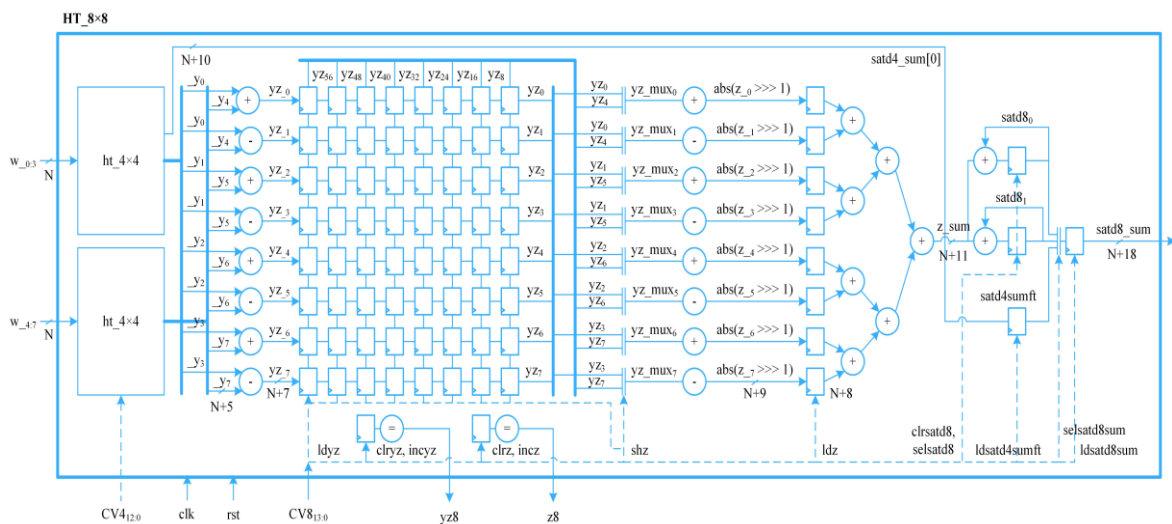


Figure 6. Proposed architecture for 8x8 2-D Hadamard transform

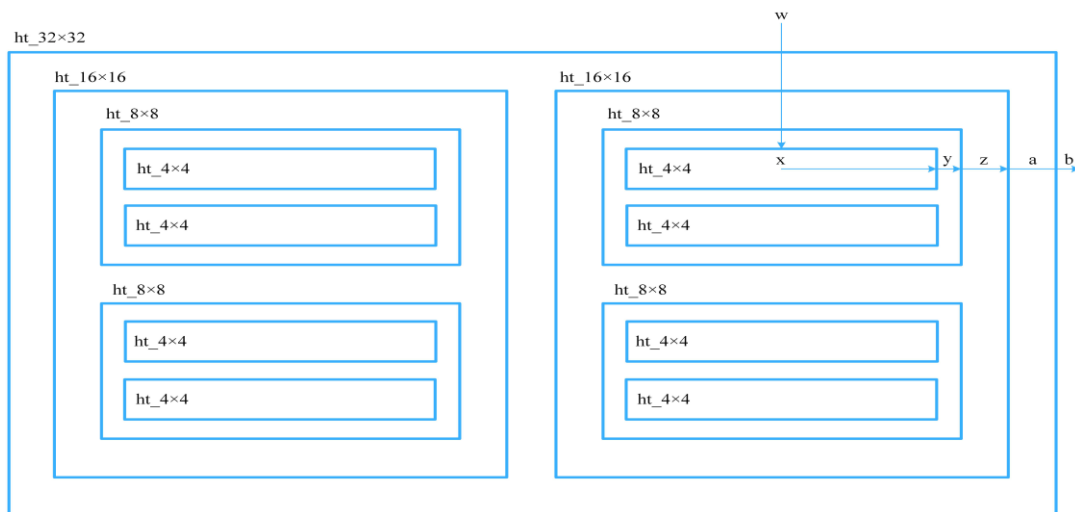


Figure 7. Simplified proposed architecture for 32x32 2-D Hadamard transform

The throughput of the designed SATD varies based on the incoming block size. 4 cycles are needed to produce the SATD sum of a 4x4 block and similarly 16 cycles are needed to produce an output from 16x16 and so on. Based on the maximum frequency in Figure 8, the throughput shown is only applicable if all incoming blocks are of the same particular size. In actual case, the percentage of the incoming block size over the total blocks has a pronounced effect on the throughput. Residual blocks of different sizes can be fed into the system and the throughout would vary between 7 million output/sec to 57 million output/sec. The throughput estimation graph is given in Figure 9, where it assumed a single block size as the continuous input.

Figure 10 shows the estimated power consumption for 4x4 to 32x32. The power consumption of 4x4 is similar to 8x8 which is around 0.486W. Starting from 16x16, the power consumption shows a slight increase of 4%, resulting in a total of 0.506 W. On the other hand, 32x32 shows the highest power consumption among all, which is 0.554 W. This is expected since the data path for a 32x32 SATD operation is much more complex compared to the others. Another interesting observation is the small power consumption increase from 4x4 to 8x8. This can be explained by the exponential increase in size from 4x4 to 32x32, therefore an exponential power increase can be observed in the overall graph.

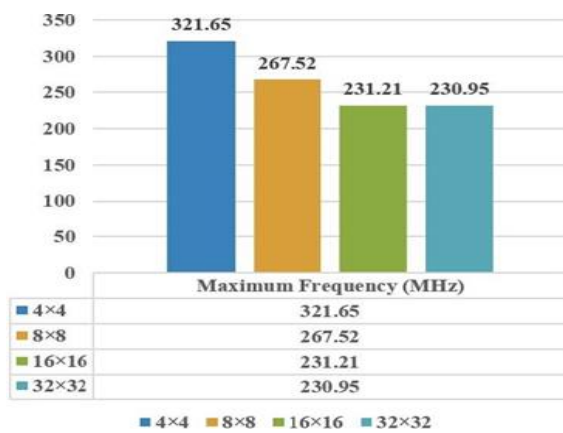


Figure 8. Maximum operating frequency of the Hadamard transform design

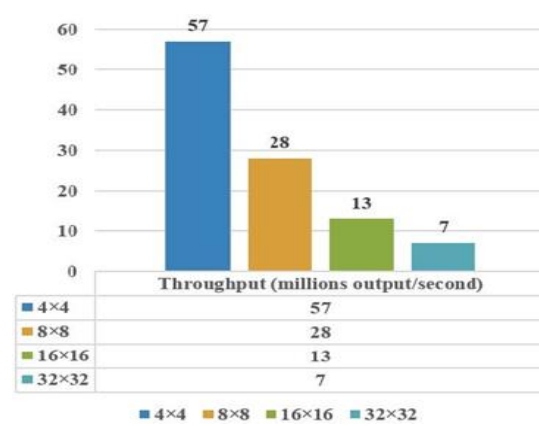


Figure 9. Throughput of the Hadamard transform design

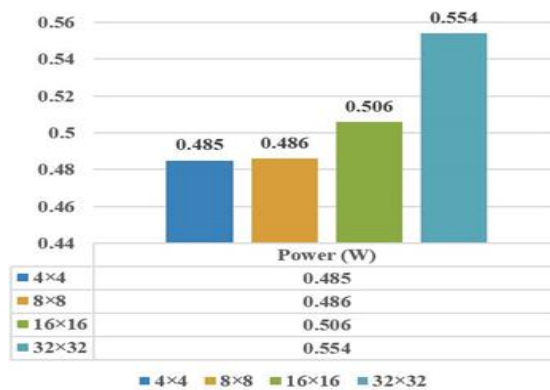


Figure 10. Power consumption of the Hadamard transform design

Most of the literature work have implemented their design onto an ASIC. As to the authors knowledge, only the work in [18] presented their results on FPGA. However, their implementation is on Altera FPGA, while our implementation is on Xilinx FPGA. The performance comparison is shown in Table 2. Based on the comparison, the utilization ratio is quite similar but our architecture can run at 267.52 MHz, which is 1.4x faster than [18] which is running at 188 MHz.

Table 2. Comparison with the Work in [18]

Metrics	[18]	Proposed work
Device	Altera Stratix III	Xilinx Kintex Ultrascale
LUT	1023	1187
FF	1486	1191
IO	Unknown	58
BUFG	Unknown	1

4. CONCLUSION

This paper describes the hardware design of the 2-dimensional Hadamard transform for Xilinx FPGA implementation. Using our proposed unified 4x4 to 32x32 architecture with pipelining and feedthrough control, variable performance has been achieved for different block sizes. Furthermore, high performance design has also been achieved compared to a similar work in literature. This custom Hadamard transform design can be used to accelerate the SATD operation in the HEVC video encoder. This is particularly useful for fast and real-time video encoding. Future work includes implementing a co-design methodology where this function is implemented on hardware, while other functions like prediction and motion vector estimation on software.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Universiti Teknologi Malaysia (UTM) for providing the facilities and funding for this research (Research University Grant (GUP) vote numbers 17H41 and 14J48). The authors would also like to acknowledge the support of the Ministry of Higher Education (MOHE) under the Fundamental Research Grant Scheme (FRGS) with the reference code FRGS/1/2016/TK04/TARUC/02/1.

REFERENCES

- [1] V. Sze, M. Budhagavi, G. J. Sullivan, High Efficiency Video Coding (HEVC) Algorithms and Architectures. Springer International Publishing, Switzerland, 2014.
- [2] Sullivan, G. J., Ohm, J.-R., Han, W.-J. and Wiegand, T. "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012. 22(12): 1649-1668.
- [3] Chen, J., Chen, Y., Karczewicz, M., Li, X., Liu, H., Zhang, L. and Zhao, X. "Coding tools investigation for next generation video coding based on HEVC," *International Society for Optics and Photonics*. 2015, vol. 9599.
- [4] Yusuf, M. S. U. and Ahmad, M. "An Efficient Inter Mode Decision Algorithm for H.264/AVC Encoding Optimization," *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*. 2015.
- [5] He, Y., Ostermann, J., Domanski, M., Au, O. C. and Ling, N. "Introduction to the Issue on Video Coding: HEVC and Beyond," *IEEE Journal of Selected Topics in Signal Processing*, 2013. 7(6): 931-933.
- [6] Lainema, J., Bossen, F., Han, W. J., Min, J. and Ugur, K. "Intra coding of the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012. 22(12): 1792-1801.
- [7] Lu, W., Yu, N., Nan, J. and Wang, D. "A Hardware Structure of HEVC Intra Prediction," *2015 2nd International Conference on Information Science and Control Engineering*. 2015.
- [8] Yi, H. and Qin, H. "The Optimization of HEVC Intra Prediction Mode Selection," *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE. 2017.
- [9] Gu, J., Tang, M., Wen, J. and Zhang, H. "A Novel SATD Based Fast Intra Prediction for HEVC," *2017 IEEE International Conference on Image Processing (ICIP)*. 2017.
- [10] L. H. Cancellier and A. B. Brscher and I. Seidel and J. L. Gntzel, "Energy-efficient Hadamard-based SATD architectures," *2014 27th Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2014, pp. 1-6.
- [11] C. Zhu and B. Xiong, "Transform-Exempted Calculation of Sum of Absolute Hadamard Transformed Differences," *IEEE Transactions on Circuits and Systems for Video Technology*, 2009, pp. 1183-1188.
- [12] Zhang, M., Liu, Y. and Liu, Z. "A New Fast Algorithm Based on SATD for HEVC Intra Prediction," *2016 Visual Communications and Image Processing (VCIP)*. 2016.
- [13] E Jaja, Z Omar, AAH Ab Rahman, M Mun'im Zabidi, "Efficient Motion Estimation Algorithms for HEVC/H. 265 Video Coding," *Information Science and Applications*, pp. 287-294
- [14] H Amer, AAH Ab Rahman, I Amer, C Lucarz, M Mattavelli, "Methodology and technique to improve throughput of fpga-based cal dataflow programs: Case study of the rvc mpeg-4 sp intra decoder," *Signal Processing Systems (SiPS), 2011 IEEE Workshop on*, pp. 186-191
- [15] Seidel, I., Beims Brascher, A., Guntzel, J. L. and Agostini, L. "Energy-efficient SATD for beyond HEVC," *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 802-805.

- [16] Dinh, V. N., Phuong, H. A., Duc, D. V., Ha, P. T. K., Tien, P. V. and Thang, N. V. "High Speed SAD Architecture for Variable Block Size Motion Estimation in HEVC encoder," *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*. 2016.
- [17] Khoo Zhi Yion, Ab Al-Hadi Ab Rahman, "Exploring the Design Space of HEVC Inverse Transforms with Dataflow Programming," *Indonesian Journal of Electrical Engineering and Computer Science*, 2017, vol. 6(1), pp. 104-109.
- [18] E. Silveira and C. Diniz and M. Beck Fonseca and E. Costa, "SATD hardware architecture based on 8x8 Hadamard Transform for HEVC encoder," *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2015, pp. 576-579.
- [19] Bryan Chan Jia Ching, Ab Al-Hadi Ab Rahman, Nabihah Ahmad, "Implementation of an 8x8 Discrete Cosine Transform on Programmable System-on-chip," *Journal of Physics: Conference Series*, 2018, Vol. 1049(1), pp. 012084.
- [20] Seidel, I., Beims Brascher, A., Guntzel, J. L. and Agostini, L. "Energy-efficient SATD for beyond HEVC," *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2016.
- [21] Liu, Z., Wang, D. and Ikenaga, T. "Hardware Optimizations of Variable Block Size Hadamard Transform for H.264/AVC FRExt," *2009 16th IEEE International Conference on Image Processing (ICIP)*. 2009.
- [22] Cancellier, L. H., Seidel, I., Brascher, A. B., Guntzel, J. L. and Agostini, L. "Exploring Optimized Hadamard Methods to Design Energy-Efficient SATD Architectures," 2015: 113-122.
- [23] Brascher, A. B., Seidel, I. and Guntzel, J. L. "Improving The Energy Efficiency of A Low-area SATD Hardware Architecture Using Fine Grain PDE," *2017 30th Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2017, pp. 155-161.
- [24] Silveira, B., Ferreira, R., Paim, G., Diniz, C. and Costa, E., "Low power SATD architecture employing multiple sizes Hadamard Transforms and adder compressors," *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2017, pp. 277-280.
- [25] Silveira, B., Abreu, B., Paim, G., Greller, M., Ferreira, R., Diniz, C., Costa, E. and Bampi, S. "Using Adder and Subtractor Compressors to Sum of Absolute Transformed Differences Architecture for Low-power Video Encoding," *2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2017, pp. 409-493.

BIOGRAPHIES OF AUTHORS



Heh Whit Ney received her bachelor degree in Electric and Electronics from Universiti Malaysia Pahang in 2016. Later in 2018, she completed Master of Computer and Microelectronic System from Universiti Teknologi Malaysia. Currently, she works as a physical design engineer in Intel Penang, Malaysia.



Ab Al-Hadi Ab Rahman received his PhD degree from the École Polytechnique Fédérale de Lausanne, Switzerland in 2013, M.Eng. degree from the Universiti Teknologi Malaysia in 2008, and B.S. degree from the University of Wisconsin-Madison, USA in 2004. His current research interests are on optimization methods and design automation for applications in video coding and deep learning. He has authored and co-authored more than 30 journals and conference papers, mainly with contributions in developing new design methodologies and techniques for high performance and low power systems. He is currently a senior lecturer at Universiti Teknologi Malaysia.



Ainy Haziyah Awab received her bachelor degree in engineering (Electrical-Electronics) from Universiti Teknologi Malaysia in 2016. Currently, she is pursuing an MPhil in Electrical Engineering (Electronic and Computer Engineering) at Universiti Teknologi Malaysia. Her current research involves video and image compression.



Mohd Shahrizal Rusli received his Ph.D in Electrical Engineering (2016), M. Eng. (Electronic and Telecommunication) (2010) and B. Eng. in Computer (2006) degree from Universiti Teknologi Malaysia (UTM), Johor. Between 2014 and 2015, he was attached to University of California Irvine, USA and University of Catania, Italy under the supervision of Prof. Dr. Nader Bagherzadeh and Assoc. Prof. Dr. Maurizio Palesi, respectively. He currently serves as senior lecturer at the Division of Electronic and Computer Engineering, UTM where his research interest and specialization are in the field of embedded processor system, deep learning application in embedded system, low-power and energy management in network-on-chip and wireless network-on-chip. He has published numerous journals, proceedings and a book chapter. He received several recognitions internationally and nationally such as best paper and national competition awards related to his research interest. He has been appointed as a reviewer for several journals. He is also an active researcher in VLSI and Embedded Computing Architecture Design (VeCAD) Research Group and has participated in several courses and seminars.



Usman Ullah Sheikh received his PhD degree (2009) in image processing and computer vision from Universiti Teknologi Malaysia. His research work is mainly on computer vision, machine learning and embedded systems design. He is currently a Senior Lecturer at Universiti Teknologi Malaysia.



Goh Kam Meng received the bachelor degree in 2010 and the Ph.D. degree in 2015 from University Teknologi Malaysia. He is currently a senior lecturer with Tunku Abdul Rahman University College in Kuala Lumpur, Malaysia. His current research interests include computer vision and recognition, artificial intelligence, and emotional information processing.