A New Multidimensional Architecture for Computing WHT

Ayman Elnaggar and Mokhtar A. Aboelaze

Abstract—**This paper presents a new recursive formulation for Walsh-Hadamard Transform (WHT) that allows the generation of higher order (longer size) multidimensional (*m*-d) WHT architectures from 2^m lower order (shorter sizes) WHT architectures. The objective of our work is to derive a unified framework and a design methodology that allows direct mapping of the proposed algorithms into modular VLSI architectures. Our methodology is based on manipulating tensor product forms so that they can be mapped directly into modular parallel architectures. The resulting WHT circuits have very simple modular structure and regular topology.**

Index Terms—**Recursive algorithms, multidimensional transforms, tensor products, WHT.**

I. INTRODUCTION

THE WALSH-HADAMARD Transform (WHT) has been used in many DSP, image, and video processing applications such as filter generating systems [1], block orthogonal transforms (BOTs) [2], and block wavelet transforms [3]. Other applications in communications are in CDMA [4] and spread spectrum [5].

This paper proposes an efficient and cost-effective methodology for mapping WHT onto VLSI structures. The main objective of this paper is to derive a design methodology and recursive formulation for the multidimensional (*m*-d) WHT which is useful for the true modularization and parallelization of the resulting computation.

The main result reported in this paper shows that a large two-dimensional WHT computation on an $n \times n$ input image can be decomposed recursively into three stages as shown in Fig. 1 for the case $n = 4$.

The second stage is constructed recursively from 2^2 parallel (data-independent) blocks each realizing a smaller-size WHT. The pre-additions and the post-permutations stages serve as "glue" circuits that combine the 2^2 lower order WHT blocks to construct the higher order WHT architecture. We also show that the proposed algorithm can be extended such that an *m*-d WHT can be constructed from 2^m smaller size *m*-d WHTs. The objective of our work is to derive a unified framework and a design methodology that allows direct mapping of the proposed algorithms into modular VLSI architectures.

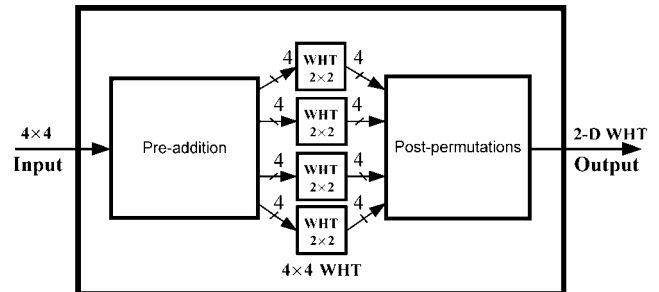Although, as far as we know from the literature, the recursive 1-d WHT algorithm is widely presented in the

Fig. 1. The proposed 2-d WHT recursive realization for a 4×4 input image.

literature [6], [7], neither the proposed *m*-d WHT algorithm nor the modular forms were previously derived.

Our work is based on a non-trivial generalization of the 1-d WHT using tensor product (or Kronecker product). It has been shown that when coupled with stride permutation matrices, tensor products provide a unifying framework for describing and programming a wide range of fast recursive algorithms for various transform [8]-[10].

Some of the tensor product properties that will be used throughout this paper are [8], [10]:

$$AB \otimes CD = (A \otimes B)(C \otimes D) \qquad (1)$$

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \qquad (2)$$

If $n = n_1 n_2$, then

$$A_{n_1} \otimes B_{n_2} = P_{n,n_1} (I_{n_2} \otimes A_{n_1}) P_{n,n_2} (I_{n_1} \otimes B_{n_2}) \qquad (3)$$

If $n = n_1 n_2 n_3$, then

$$I_{n_1} \otimes A_{n_2} \otimes I_{n_3} = P_{n,n_1 n_2} (I_{n_1 n_3} \otimes A_{n_2}) P_{n,n_3} \qquad (4)$$

If $2n = n_1 n_2$, then

$$P_{n,2} = P_{n,n_1} P_{n,n_2} \quad , \qquad (5)$$

where $\otimes$ denotes the tensor product, I_n is the identity matrix of size n, and $P_{n,s}$, the permutation matrix, is $n \times n$ binary matrix whose entries are zeroes and ones, such that each row or column of this matrix has a single 1 entry. If $n = rs$ then $P_{n,s}$ is an $n \times n$ binary matrix specifying an (n/s)-shuffle (or s-stride) permutation. The effect of the permutation matrix $P_{n,s}$ on an input vector X_n of length n is to shuffle the elements of X_n by grouping all the r elements separated by distance s together. The first r elements will be $x_0, x_s, x_{2s}, \cdots, x_{(r-1)s}$, the next r elements are $x_1, x_{1+s}, x_{1+2s}, \cdots, x_{1+(r-1)s}$, and so on.

II. THE MODIFIED FORMULATIONS OF THE 1-D WHT

In this Section, we modify the original 1-d WHT to the iterative form that allows a hardware saving without affecting the processing speed.
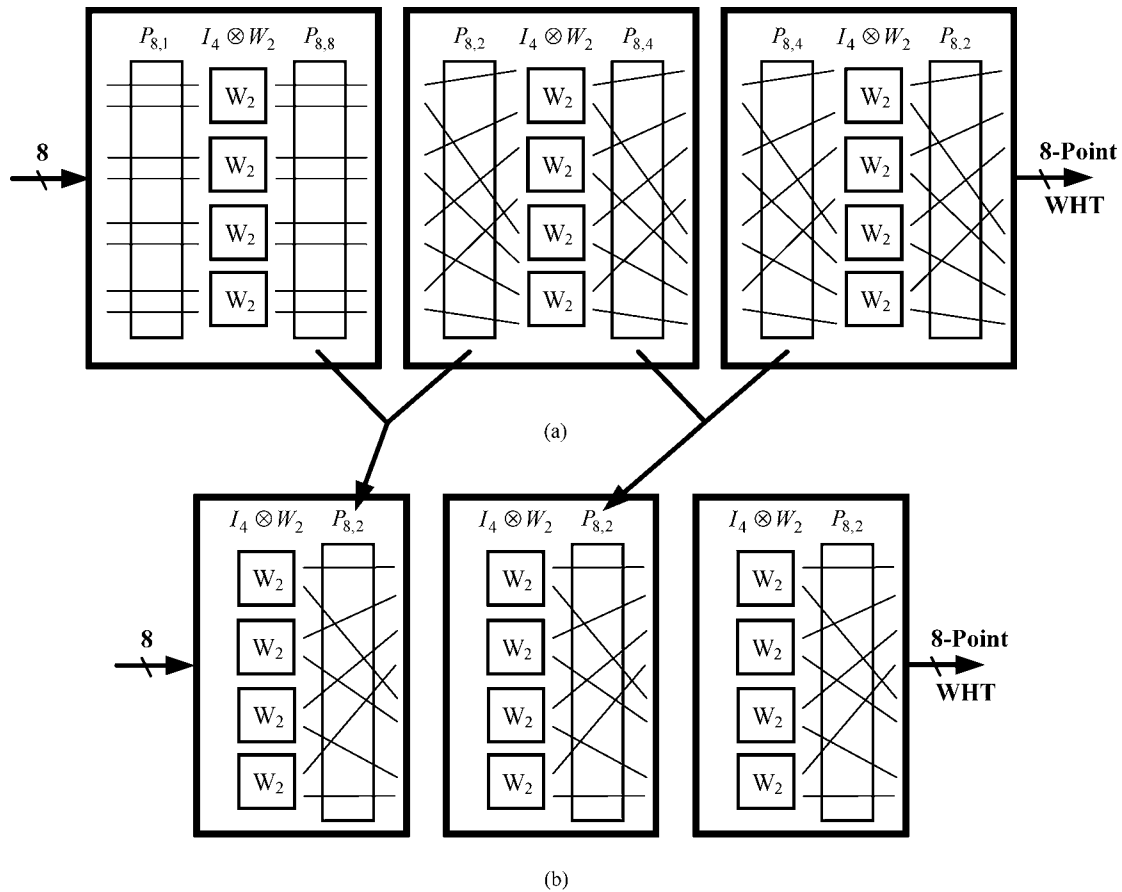
www.SID.ir

(a)

(b)

Fig. 2. The realization of (a) the original 1-d WHT iterative algorithm and (b) the modified 1-d WHT algorithm.

The original 1-d WHT transform matrix is defined as [6]

$$W_n = \begin{pmatrix} W_{n/2} & W_{n/2} \\ W_{n/2} & -W_{n/2} \end{pmatrix}, \quad W_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{6}$$

where W_2 is the 2-point WHT.

A. The 1-d WHT Iterative Formulation

Let $k = \log_2 n$, we can write (6) in the iterative tensor-product form

$$W_n = W_2 \otimes W_{n/2} = W_2 \otimes W_2 \otimes \cdots \otimes \cdots W_2$$
$$= \prod_{i=0}^{k-1} (I_{2^i} \otimes W_2 \otimes I_{2^{k-i-1}}) \tag{7}$$

which using property (4), can be modified to

$$W_n = \prod_{i=0}^{k-1} P_{n,2^{i+1}} (I_{2^{k-1}} \otimes W_2) P_{n,2^{k-i-1}} . \tag{8}$$

As an example, we can express W_8 as

$$W_8 = [P_{8,2}(I_4 \otimes W_2)P_{8,4}][P_{8,4}(I_4 \otimes W_2)P_{8,2}].$$
$$[P_{8,8}(I_4 \otimes W_2)P_{8,1}] \tag{9}$$

The realization of W_8 is shown in Fig. 2(a).

Applying property (5) to (9) and noting that now the permutations in two adjacent stages can be grouped together into a single permutation, the adjacent permutations $P_{8,2} P_{8,8}$ (from the first and the second stage) will be replaced by the single permutation $P_{8,2}$ and the adjacent permutations $P_{8,4} P_{8,4}$ (from the second and the third stage) will be replaced by the single permutation $P_{8,2}$ as shown in Fig. 2(b).

Similarly, (8) can be simplified to

$$W_n = \prod_{i=0}^{k-1} P_{n,2} (I_{2^{k-1}} \otimes W_2) . \tag{10}$$

Thus, W_n can be computed by the cascaded product of k similar stages (independent of i) of double matrix products instead of the triple matrix products in (8). Alternatively, we can realize (10) by a single block of $P_{n,2} (I_{2^{k-1}} \otimes W_2)$ and take the output after k iterations that allows a hardware saving without slowing down the processing speed and reduction in the hardware size as shown in Fig. 3 for the case $n = 8$.

It should be mentioned that we have applied property (5) to reduce the shuffling inherited in the original WHT algorithm to allow a uniform hardware blocks as shown in Fig. 2 (b). We have not modified the original complexity of the WHT that are centered in the W_2 blocks as shown in Figs. 2 and 3.

B. The 1-d WHT Recursive Formulation

Applying property (1), (7) can be modified to

$$W_n = W_2 \otimes W_{n/2} = I_2 W_2 \otimes W_{n/2} I_{n/2}$$
$$= (I_2 \otimes W_{n/2})(W_2 \otimes I_{n/2}) \tag{11}$$
$$= (I_2 \otimes W_{n/2})Q_n$$

$$\tag{11}$$

where,

$$Q_n = (W_2 \otimes I_{n/2}) . \tag{12}$$

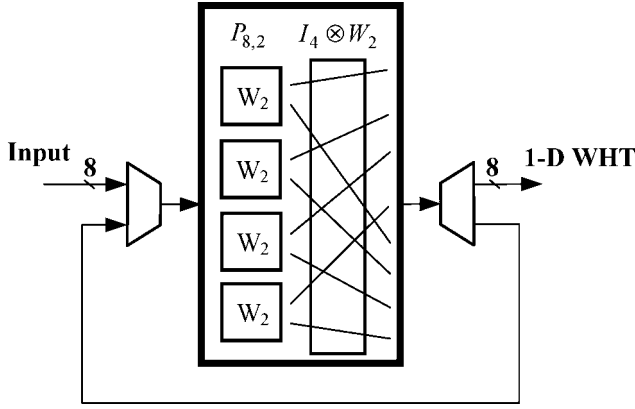Equation (11) represents the two-stage recursive tensor product formulation of the 1-d WHT in which the first stage

Fig. 3. The reduced hardware realization of the modified 1-d WHT algorithm.

Fig. 4. The realization of the recursive 1-d WHT.

is the pre-additions (Q_n), followed by the second stage of the core computation ($I_2 \otimes W_{n/2}$) that consists of a parallel stage of two identical smaller WHT computations each of size $n/2$ as shown in Fig. 4.

III. THE PROPOSED FORMULATION OF THE 2-D WHT

This section will develop two recursive methodologies for realizing the 2-d WHT computations from smaller WHT computations. First, we present a direct method for realizing the 2-d WHT computations using the conventional row-column decomposition of the 1-d WHT in a tensor product form. The second methodology provides a truly 2-d recursive structures that employ one stage of smaller 2-d WHTs to realize the large 2-d WHT.

A. Conventional Row-Column Decomposition

Since the WHT matrix is separable [11], the 2-d WHT for an input image of dimension $n_1 \times n_2$ can be computed by a stage of n_2 parallel 1-d WHT computations on n_1 points each, followed by another stage of n_1 parallel 1-d WHT computations on n_2 points each. This can be represented by the matrix-vector form,

$$X = W_{n_1,n_2} x ,\qquad (13)$$

where W_{n_1,n_2} is the 2-d WHT transform matrix for an $n_1 \times n_2$ image, X and x are the output and input column-scanned vectors, respectively. For separable transforms, the matrix W_{n_1,n_2} can be represented by the tensor product form [11],

$$W_{n_{1,n_2}} = W_{n_1} \otimes W_{n_2} \qquad (14)$$

where W_{n_1} and W_{n_2} are the row and column 1-d WHT operators, as defined by (7), on x, respectively.

By substituting (14) in (13), we have

$$X = (W_{n_1} \otimes W_{n_2})\, x , \qquad (15)$$

which using (7) can be expressed as

$$X = W_{n_1 \times n_2} x . \qquad (16)$$

Therefore, the 2-d WHT on an $n_1 \times n_2$ input is equivalent to a 1-d WHT on a 1-d input vector of size $n_1 \times n_2$ that can be implemented using either the modified 1-d iterative algorithm given by (10) or the modified 1-d recursive algorithm given by (11).
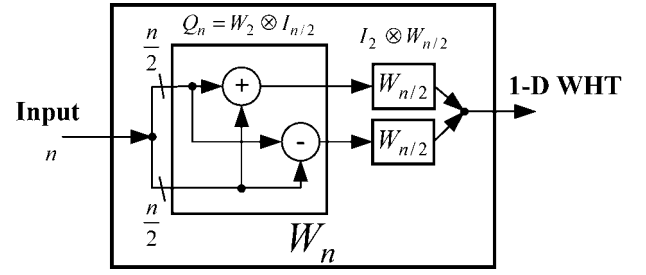
B. The Truly Recursive Formulation of the 2-D WHT

Now we will derive a truly 2-d recursive formulation of the WHT by further manipulation of (15). Substituting (11) in (15), the 2-d WHT transform matrix can be written as,

$$W_{n_1,n_2} = [(I_2 \otimes W_{n_1/2})\, Q_{n_1}] \otimes [(I_2 \otimes W_{n_2/2})\, Q_{n_2}] . (17)$$

Applying property (1), we can write W_{n_1,n_2} as

$$\begin{aligned} W_{n_1,n_2} &= [(I_2 \otimes W_{n_1/2}) \otimes (I_2 \otimes W_{n_2/2})] \\ &\quad [Q_{n_1} \otimes Q_{n_2}] \\ &= [C_{n_1,n_2}\, Q_{n_1,n_2}] \end{aligned} \qquad (18)$$

here

$$\begin{aligned} C_{n_1,n_2} &= [(I_2 \otimes W_{n_1/2}) \otimes (I_2 \otimes W_{n_2/2})], \\ Q_{n_1,n_2} &= [Q_{n_1} \otimes Q_{n_2}] \end{aligned} \qquad (19)$$

Now, from property (2), we can write C_{n_1,n_2} in the form,

$$C_{n_1,n_2} = (I_2 \otimes W_{n_1/2} \otimes I_2) \otimes W_{n_2/2} \qquad . \qquad (20)$$

Applying property (4), we can write (20) in the form

$$\begin{aligned} C_{n_1,n_2} &= [P_{2n_1,n_1}(I_4 \otimes W_{n_1/2})\, P_{2n_1,2}] \otimes [W_{n_2/2}] \\ &= [P_{2n_1,n_1}(I_4 \otimes W_{n_1/2})\, P_{2n_1,2}] \otimes \\ &\quad [I_{n_2/2}\, W_{n_2/2}\, I_{n_2/2}] \\ &= [P_{2n_1,n_1} \otimes I_{n_2/2}][(I_4 \otimes W_{n_1/2}) \otimes W_{n_2/2}] \\ &\quad [P_{2n_1,2} \otimes I_{n_2/2}] \\ &= [P_{2n_1,n_1} \otimes I_{n_2/2}][(I_4 \otimes W_{n_1/2,n_2/2}] \\ &\quad [P_{2n_1,2} \otimes I_{n_2/2}] \end{aligned} \qquad (21)$$

where

$$W_{n_1/2,\,n_2/2} = W_{n_1/2} \otimes W_{n_2/2} . \qquad (22)$$

Finally, substituting (21) in (18), we have

$$W_{n_1,n_2} = [\widetilde{R}_{n_1,n_2}\,(I_4 \otimes W_{n_1/2,n_2/2})\,\widetilde{Q}_{n_1,n_2}] \qquad (23)$$

where

$$\begin{aligned} \widetilde{Q}_{n_1,n_2} &= (P_{2n_1,2} \otimes I_{n_2/2})\, Q_{n_1,n_2} , \\ \widetilde{R}_{n_1,n_2} &= (P_{2n_1,n_1} \otimes I_{n_2/2}) . \end{aligned} \qquad (24)$$

Equation (23) represents the truly recursive 2-d WHT in which $\widetilde{Q}_{n_1,n_2}$ and $\widetilde{R}_{n_1,n_2}$ are the pre- and post-processing glue structures, respectively, that combine 2^2 identical lower-order 2-d WHT modules each of size $n_1/2 \times n_2/2$ in parallel, to construct the higher order 2-d WHT of size $n_1 \times n_2$ as shown previously in Fig. 1.

IV. THE MULTI-DIMENSIONAL WHT

We can extend the 2-d WHT derivation proposed in Section III to the m-d case. From (14), the m-d WHT can be written in the tensor product form

$$X = (W_{n_1} \otimes W_{n_2} \otimes \cdots \otimes W_{n_m})\, x$$
$$= (\overset{m}{\underset{i=1}{\otimes}} W_{n_i})\, x \tag{25}$$

where, $(W_{n_1} \otimes W_{n_2} \otimes \cdots \otimes W_{n_m})$ is the m-d WHT transform matrix for an m-d input, W_{n_i} is the 1-d WHT coefficient matrix for an input vector of length n_i as defined in (7), X and x are the output and input column-scanned vectors, respectively.

Following the steps in Section III, we can write (25) in the form

$$X = [\, \hat{R}\, (I_{2^m} \overset{m}{\underset{i=1}{\otimes}} W_{n_i/2})\, \hat{Q}]\; x \quad , \tag{26}$$

where $\overset{m}{\underset{i=1}{\otimes}} W_{n_i/2}$ is the lower order m-d WHT and

$$\hat{Q} = \left(\prod_{i=1}^{m-1} \left(P_{u_1,u_2} \overset{i}{\underset{k=1}{\otimes}} I_{u_3} \right) \right) \left(\overset{m}{\underset{i=1}{\otimes}} Q_i \right) \quad ,$$

$$\hat{R} = \prod_{i=1}^{m-1} \left(P_{w_1,w_2} \overset{m-1}{\underset{l=i}{\otimes}} I_{w_3} \right) \quad ,$$

$$u_1 = 2 \prod_{j=1}^{m-i} n_j \quad , \quad u_2 = 2 \, , \quad u_3 = \frac{1}{2} \prod_{j=1}^{i} (n_{m-j+1}),$$

$$w_1 = \frac{1}{2} \prod_{k=1}^{i} (n_k), \; w_2 = \prod_{k=1}^{i} (n_k), \; w_3 = \frac{1}{2} \prod_{j=i+1}^{i} (n_j), \tag{27}$$

Q_i is the 1-d pre-processing as defined by (12).

Equation (26) extends our results by showing that a large m-d WHT can be computed from a single stage of smaller m-d WHTs.

V. CONCLUSIONS

In this paper, we proposed a new recursive formulation for Walsh-Hadamard Transform (WHT) that allows the generation of higher order (longer size) multidimensional (m-d) WHT architectures from 2^m lower order (shorter sizes) WHT architectures in a parallel realization with the addition of two glue stages of pre-additions and post-permutations. The resulting networks have very simple modular structures and highly regular topologies.

The merits of the proposed architectures can be summarized as:

1. They provide a recursive design for WHT. The same design can be used if the size of the problem changes (i.e. the number of the inputs is increased for example).

2. User can freely choose a very fast WHT architecture as a core design and recursively build the complete architecture using only additional modular stages of pre-additions and post-permutations which make it well suited for automated synthesis.

3. As shown in Fig. 1, there is no communication between the parallel core blocks and hence, very high speed is achievable due to low communication overhead. Moreover, the architecture is easy to partition among multiple chips.

4. Fully pipelined architecture that allows for full utilization of the architecture. Moreover, the area can be reduced dramatically with a scheduled pipeline.

REFERENCES

[1] S. Samadi, A. Nishihara, and H. Iwakura, "Filter generating systems," *IEEE Trans. on Circuits and Systems II,* vol. 47, no. 8, pp. 214-221, Mar. 2000.

[2] Makur, "BOT's based on non-uniform filter banks," *IEEE Trans. on Signal Processing,* vol. 44, no. 8, pp. 1971-1981, Aug. 1996.

[3] E. Cetin, O. N. Gerek, and S. Ulukus, "Block Wavelet transforms for image coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 6, pp. 433-435, Dec. 1993.

[4] F. Adachi, K. Ohno, A. Higashi, T. Dohi, and Y. Okumura, "Coherent multicode DS-CDMA mobile radio access," *IEICE Trans. on Communications,* vol. E79-B, no. 9, pp. 1316-1325, Sep. 1996.

[5] R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to Spread Spectrum Communications*, Prentice Hall, 1995.

[6] S. Rahardja and B. J. Falkowski, "Family of unified complex Hadamard transforms," *IEEE Trans. on Circuits and Systems II*, vol. 46, no. 8, pp. 1094-1100, Aug. 1999.

[7] R. K. E. Yarlagadda and J. E. Hershey, *Hadamard Matrix Analysis and Synthesis with Applications to Communications and Signal/Image Processing*, Kluwer Academic Publishers, Boston, 1997.

[8] Elnaggar and M. A. Aboelaze, "An efficient architecture for multi-dimensional convolution," *IEEE Trans. on Circuits and Systems II*, vol. 47, no. 12, pp. 1520-1523, Dec. 2000.

[9] Elnaggar, M. A. Aboelaze, and A. Al-Naamany, "A modified shuffle-free Architecture for linear convolution," *IEEE Trans. on Circuits and Systems II*, vol. 48, no. 9, pp. 862-866, Sep. 2001.

[10] R. Tolimieri, M. An, and C. Lu, *Algorithms for Discrete Fourier Transform and Convolution*, Springer-Verlag, New York, 1989.

[11] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons Inc., 1991.

Ayman I. Elnaggar received the B.Sc. from Cairo University, Cairo Egypt in 1984, and the Ph.D. degree from University of British Columbia, Vancouver in 1997 in Computer Engineering.

During the period 1996-1997, he worked as an ASIC Designer with Nortel, Ottawa, and with PMC-Sierra, Vancouver. Dr. Elnaggar taught many undergraduate and post-graduate courses, and presented short courses in Canada, UAE, Oman in several Computer Engineering areas.

He is the recipient of the GREAT Award 1996-1997 from the BC Science Council, Canada. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Sultan Qaboos University, Oman. His research interests include networked multimedia, e-learning, multimedia systems, and networking of smart buildings.

Dr. Elnaggar is a member of the Institute of Electrical and Electronics Engineers (IEEE) and a member of the Association For Computing Machinery (ACM).

Mokhtar A. Aboelaze received the B.Sc. from Cairo University, Cairo Egypt in 1978, M.Sc. from the University of South Carolina, Columbia S.C. in 1984, and the Ph.D. degree From Purdue University in W. Lafayette IN. in 1988 all in Electrical and Computer Engineering.

Dr. Aboelaze joined the Department of Computer Science at York University in Toronto Canada in 1988, where he is now an Associate Professor. His research interests include computer networks, wireless networks, medium access protocols in wireless networks, computer architecture, and special purpose architecture.

Dr. Aboelaze is a member of the Institute of Electrical and Electronics Engineers (IEEE) and a member of the Association of Professional Engineers of Ontario, Canada.