Memoryless Viterbi Decoder

Dalia A. El-Dib, *Member, IEEE,* M.I. Elmasry, *Fellow, IEEE,*

***Abstract*— The problem of survival memory management of a Viterbi Decoder (VD) was solved by introducing a novel pointer implementation for the register exchange (RE) method, where a pointer is assigned to each row of memory in the SMU [1]. The content of the pointer which points to one row of memory is altered to point to another row of memory, instead of copying the contents of the first row to the second.**

In this paper, the one-pointer VD is proposed; if the initial state of the convolutional encoder is known, the entire survivor memory unit (SMU) is reduced to only one row. Because the decoded data are generated in the required order, even this row of memory is dispensable. Thus, the one-pointer architecture, referred to as memoryless Viterbi Decoder (MLVD), reduces the power consumption of a traditional trace back (TB) VD by approximately 50 percent. A prototype of the MLVD with a one third convolutional code rate and a constraint length of nine is mapped into a Xilinx 2V6000 chip, operating at 25 MHz with a decoding throughput of more than 3Mbps and a latency of two data bits.

***Index Terms*— wireless, low power, Viterbi Decoder, register exchange, memoryless.**

I. Introduction

THE, Viterbi Algorithm is an efficient method for the realization of maximum likelihood decoding of convolutional codes [2] [3]. The digital VD is widely used in many digital wireline and wireless applications. Existing VDs are categorized by the way how the decoded data bits are stored in and retrieved from the SMU. Two methods are mainly used, the RE and the TB method [4]. In the literature, the RE method is acceptable for trellises with only a small number of states, whereas the TB approach is acceptable for trellises with a large number of states. Therefore, the TB method has been widely investigated and implemented. For example, an M-layered approach that combines the M-stages of the trellis into one stage has been proposed [5]. Bit-serial approaches and operation reformulations have also been initiated [6] [7]. Although several attempts to reduce the power consumption of the TB VD have been proposed, only a few attempts to combine the advantages of the TB and RE methods have been reported. The RE method has been modified by Han et al [8] to reduce its memory access rate. Their modified RE (MRE) method eliminates the trace-back operation and reduces the amount of memory. The MRE can be designed with emphasis on either efficient memory or low latency, but no estimated power reduction figures were given. All of the previous design approaches were developed for low constraint length VDs (K=3 to K=7). The decoders in [9], [10], and [11] were designed for CDMA applications for which the constraint length must be K=9. Although Kang and Willson have introduced a very low power TB VD [10], its speed is limited due to the use of sequential architectures for the ACS processing. The decoder that was devised by Chang et al. has an even lower power consumption and achieves speeds in the range of $Mbps$ [11].

The modified RE method proposed in [1] suggests a variation to the Viterbi Algorithm (VA), and has an estimated power reduction of 20 percent over the conventional low power VD proposed by Chang et al, but has considerable performance degradation, which was accidently overlooked in [1]. Other variations to the VD have also been proposed in the literature. Both the T-algorithm [12] and the M-algorithm [13] apply some techniques to reduce the number of paths being traced back. Some adaptive reduced-complexity techniques have been applied to these suboptimal VAs. The adaptive approaches perform as decent as conventional VDs with a significant reduction in the computational complexity if the proper system parameters have been selected [14]. None of these algorithms ensure the amount of reduction in the computational effort.

The modified RE method [1], which is further improved in this work, utilizes the pointer concept, that is widely used in software engineering. Instead of moving the contents of one row of memory to a second row of memory, the pointer to the first row is altered to point to the second row. The pointer to one row of memory simply carries the current state in the trellis of the VD. The pointer implementation avoids the power hungry register exchange operations of the traditional RE method, and is referred to as pointer Viterbi Decoder (PVD). In the next section the PVD is briefly reviewed, then its memoryless version is introduced.

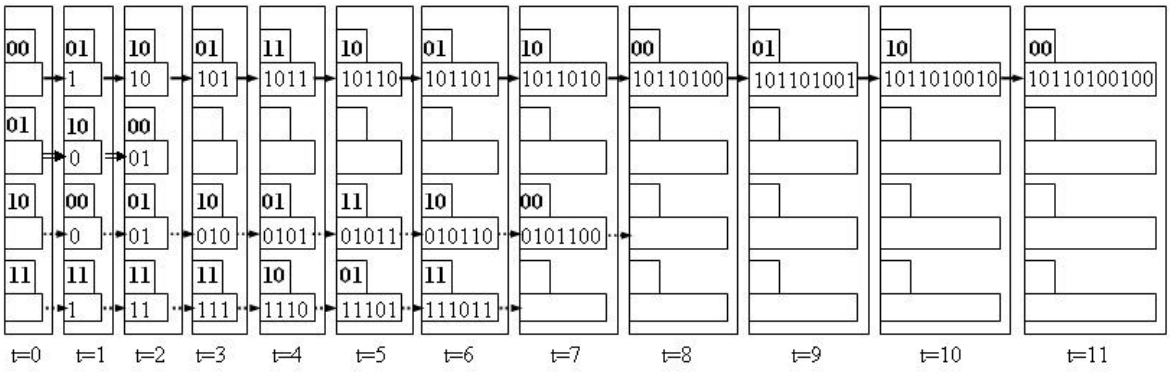
II. The Memoryless Viterbi Decoder (MLVD)

The PVD keeps track of the current row position of the decoder in the memory. It makes use of the fact that the bit appended to each row of memory is exactly the bit that is shifted into the pointer to form the new pointer to that row of memory. To show the functionality a 4-state rate=1/3 convolutional encoder (G0=101, G1=111, and G2=111) is employed to encode the input sequence of (10110100100). The code stream, (111, 011, 000, 100, 100, 000, 011, 111, 111, 011, 111) is generated and transmitted over a noisy channel. The noisy code stream, (111, 011, 00<u>1</u>, 100, 100, 000, 011, 111, 11<u>0</u>, 011, 111), for example, is received at the decoder. The underlined bits are incorrect because of the noise encountered during transmission. Applying the modified RE method results in the diagram illustrated in Figure 1. This figure displays the successive values for the pointers and rows of memory over time.

Manuscript received July, 2004; This work was supported by the Egyptian Governement.

Dalia A. El-Dib is with the Electronic Research Institute, Cairo, Egypt (email: dafeldib@eri.sci.eg) and M.I. Elmasry is with the University of Waterloo, Ontario, Canada.

Fig. 1. New RE approach with pointer implementation (the upper register carries the pointer and the lower register carries the decoded bits)



A closer look reveals that each row of memory is used to trace the decoded bits, if an initial state is assumed. The first row of memory decodes the data, if an initial state, S_0, is assumed. The last row records the decoded data, if an initial state, S_{255}, is assumed, and so on. If the initial state is known, are all these rows of memory necessary? Absolutely not. For example, if the initial state is zero, then only the first row of memory is needed. In other words, the storage of the decoded bits is necessary in order to choose only one row of memory at the end to represent the actual decoded bits. If the required row of memory is predetermined, then there is no need for the storage of the other rows. Furthermore, there is no need for the storage of the row that is assigned to the predetermined initial state, because the RE approach generates the decoded bits in the correct order. The decoded bits are produced, and then read out from the decoder. Thus, a memory-free VD can be implemented by solely resetting the encoder contents for each L bits that are encoded, where L is the survivor path length (5*K). There is no need to interrupt the data sequence nor to transmit a long sequence of zero data bits. The encoded data are continuous, but the contents of the encoder bits (eight bits for $K = 9$ convolutional encoder) are reset to zero for each L bits transmitted. Whereas the pointer of the VD in the receiver needs to be reset to the initial state for each L bits decoded. Besides, the memory-free VD has a latency of only two data bits, whereas the latency of a conventional VD is at least L bits long. The new memory-free VD implementation is called, the Memoryless Viterbi Decoder (MLVD). Since the MLVD needs to track only one row, the MLVD requires only one pointer to track the current position of the decoder in the trellis in Figure 2. The MLVD is designed in VHDL for WCDMA applications with the specifications listed in Table I.

TABLE I
VD SPECIFICATIONS

Constraint Length	K = 9
Coding Rate	r=1/3
Generator Polynomials	$G0$= 557, $G1$=663 $G2$=711
Decision Level	3-bit Soft Decision
Target Speed	$2Mbps$

III. ARCHITECTURE OF THE MLVD

The MLVD is an extra low power design for a VD with the only restriction of resetting the encoder register at each L of the encoded data bits and providing the necessary synchronization with the decoder. The block diagram of the MLVD, designed in VHDL, is shown in Figure 3.

In order to have a built-in self-test design, a Linear Feedback Shift Register (LFSR) and a comparator are added. The LFSR produces the random input for the encoder, whereas the comparator compares the delayed version of the LFSR's output with the decoded output of the MLVD. An output signal, status, indicates the correct functionality of the design. The following is a discussion of the different parts of the MLVD design and their functionality.

A. Convolutional Encoder

The convolutional encoder that is implemented is that of the reverse link for WCDMA applications. It is a K=9 and r=1/3 convolutional encoder. To implement a 3-bit soft-decision VD, the output of the encoder is translated from (0,1) to (111, 011). 111 is the two's complement representation of the decimal number -3, and 011 is the representation of the decimal number 3. The output of the encoder is fed directly (without noise) into the first block of the VD, the BMU.

B. Branch Metric Unit (BMU)

For binary convolutional codes, it is proven that linear distances (Hamming distances) can be used as the optimum branch metrics (BMs) [15]. For three 3-bit soft decision input bits, (i_0, i_1, i_2), each ranging from -3 to 3, eight 5-bit BMs are generated. The BMU performs simple add and subtract operations on the decision bits to generate the output as represented in Figure 4, and the output of the BMU is still in a two's complement format. The bit serial format of the BMs is generated by the parallel to serial module at the output of the BMU, as shown in Figure 3, then the bit serial format $BM < 0 : 7 >$ is fed into the ACSU.

C. Add Compare Select Unit (ACSU)

The ACSU is composed of 126 units; each is composed of an ACS butterfly module, which adds the BMs to the

Fig. 2. MLVD approach with pointer implementation (the upper box carries the pointer and the lower box carries the decoded bits stored in memory)

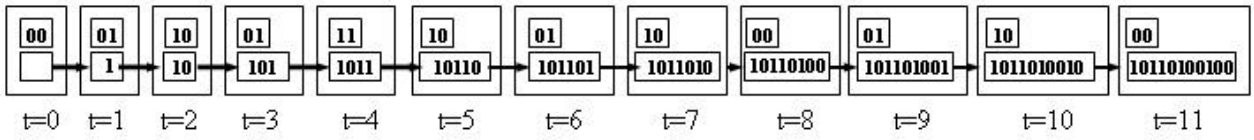


Fig. 3. MLVD block diagram

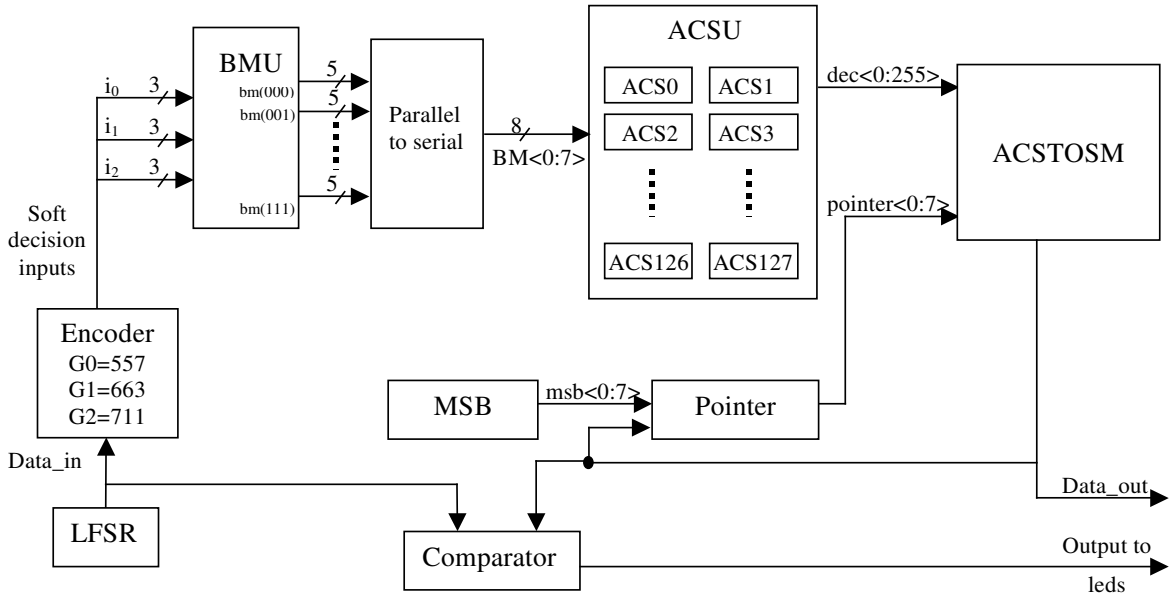
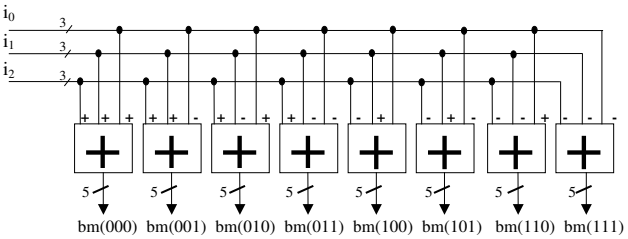


Fig. 4. Branch Metric Unit (BMU) operations



corresponding path metrics (PMs), compares the new PMs, and then feeds the selected PMs back into the ACSU. The typical operation of one ACS butterfly module is as follows:

If $BM^{(i,p)} + PM^{(i,p)} < BM^{(j,p)} + PM^{(j,p)}$
Then $Dec^{(p)} = 0$
$PM^{(p)} = PM^{(i)} + BM^{(i,p)}$

If $BM^{(i,p)} + PM^{(i,p)} > BM^{(j,p)} + PM^{(j,p)}$
Then $Dec^{(p)} = 1$
$PM^{(p)} = PM^{(j)} + BM^{(j,p)}$

If $BM^{(i,q)} + PM^{(i,q)} < BM^{(j,q)} + PM^{(j,q)}$
Then $Dec^{(q)} = 0$
$PM^{(q)} = PM^{i} + BM^{(i,q)}$

If $BM^{(i)} + PM^{(i,q)} > BM^{(j)} + PM^{(j,q)}$
Then $Dec^{(q)} = 1$
$PM^{(q)} = PM^{(j)} + BM^{(j,q)}$.

And according to the symmetric characteristics of the VD (Table I),

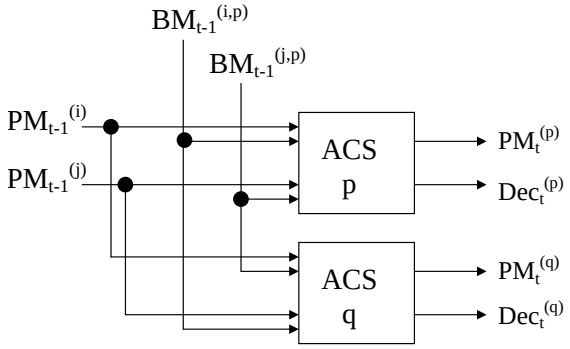
$BM^{(i,q)} = BM^{(j,p)}$
and $BM^{(i,p)} = BM^{(j,q)}$.

Therefore, only two branch metrics BMs are connected to each butterfly unit as shown in Figure 5. The bit serial approach proposed by Chang et al. [11] is adopted for the MLVD implementation to reduce the routing overhead among the different ACSU modules.

D. Add Compare Select TO Survivor Memory (ACSTOSM)

The ACSTOSM is employed to route the decision of the appropriate ACS module to the output. The ACSTOSM is a 256 to one decoder. The select signal for this large decoder is the output of the pointer module. The output of the ACSTOSM module is already the decoded output sequence of the MLVD,

Fig. 5. One ACS butterfly module



but is also fed back into the pointer module to update the current state in the decoding trellis.

E. Pointer

The pointer block contains the current state of the decoder (eight bits). It is reset to zero (the initial state of the encoder) for each L of bits decoded. Then, for each bit decoded, the pointer content is updated, by the output of the ACSTOSM module. The exact position of the bit that will be updated is determined by the MSB block, which acts as a circular pointer to the pointer block.

IV. Power Consumption and Performance of the MLVD

To calculate the power reduction estimation, cost values for the operations required by the SMU of the TB VD [11], the PVD, and the MLVD are provided in Table II. It is noted that many SMU's operations are no longer executed with the MLVD.

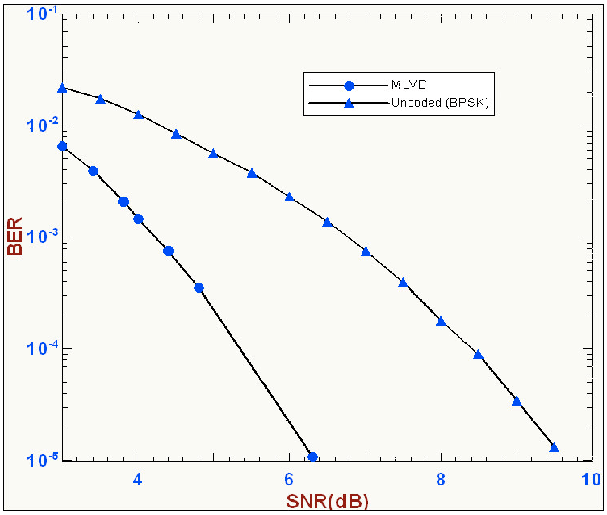
TABLE II
ESTIMATED COST FUNCTION TO DECODE 48 CODEWORDS

Operation	TB	PVD	MLVD
Writing decision bits into the memory	256 x 48 12288w	(256 x 48)/2 6144w	–
Reading from the memory	48 x 3 144r	48 48r	–
Writing into the MSB of the pointers	–	256 x 48 12288rg	48 48rg
Writing termination bits into the termination registers	–	256 256rg	–
8-bit register shifting (shift to the right and append a bit to the LSB)	48 x 3 x 8 1152rg	–	–
Total	12504w	7736w	6w

W, r, and rg represent the power dissipation cost function for writing one bit into the memory block, reading one bit from the memory block, and writing one bit into a register, respectively. As estimated in [1], w:rg $\sim$ 8:1 and w:r $\sim$ 2:1. If the equivalent estimated value of w is substituted for r and rg, then the total in Table II is produced. The total shows that there is almost no power consumption that is associated with the SMU of the MLVD. This occurs because all the SMU operations are related to the memory, and for the MLVD, no memory is required. Typically, the SMU consumes about 50 percent of the total VD power [16]. Thus the MLVD reduces the power consumption of the VD by half.

A Java program is written to simulate the MLVD performance. The data signal of the convolutional encoder ($K = 9, r = 1/3, L = 48$) is converted into antipodal representation, then additive white gaussian noise (AWGN) is added, then the resulting signal is quantized into 3-bit soft representation and is input to the MLVD. The coding gain of the MLVD is plotted in Figure 6. The MLVD achieves a coding gain of 2.6 dB at a $SNR(E_b/N_0) = 10^{-3}$; the coding gain is increased to 3.3 dB at a SNR of 10^{-5}.

Fig. 6. Coding gain simulations of the MLVD ($K = 9, r = 1/3, L = 48$).



V. Xilinx Implementation and Test Results

To prepare the MLVD's VHDL design that is implemented on the FPGA, the design is synthesized by using Synopsys tools. Then, the design is imported into Xilinx tools for the mapping and routing; then a VHDL file with timing information is re-simulated for the timing verification. Afterward, the MLVD's design is downloaded to a 2V6000 Xilinx chip, which is mounted on a rapid prototyping system including the LT-XC2V6000 logic tile. The logic tile provides some leds to be connected to the output of the Xilinx Chip. These leds are used to flash serially in the case of the correct functionality. The MLVD consumes only 8 percent of the total slices of the 2V6000, comprising a total of 68,736 gates. The design consumes 16 I/O bits, and can be implemented on a much smaller FPGA, but was not available. Even the power consumption of the design on the FPGA is not very significant, because the main objective is to test the design feasibility and operability. The implemented design on the Xilinx FAPGA operates at 25 MHz with a decoding throughput of more than 3 $Mbps$. Thus the target speed of 2 $Mbps$ (see Table I) is achieved.

VI. Conclusion

The PVD proposed by the authors in a previous paper [1] was further improved to reduce its power consumption. The MLVD is a memoryless implementation of the VA, and successfully decodes the continuous data encoded by a WCDMA convolutional encoder. The power reduction is as high as 50 percent and the latency is only 2 data bits. The new implementation is realized by applying the pointer concept to the RE implementation, and by reinforcing the initial state of the convolutional encoder every L bits encoded. The BER rate for the MLVD is estimated to be 10^{-3} for an SNR of 4.2dB with a coding gain of 2.6dB. The MLVD along with an on-chip convolutional encoder (rate=1/3, K=9) was implemented on a Xilinx 2V6000 chip to demonstrate both the design's functionality and feasible implementation. The hardware and computational overhead of the new implementation is only a 256 to 1 decoder, which is switching at the data rate frequency.

References

[1] D. A. El-Dib and M. I. Elmasry, "Modified register-exchange viterbi decoder for low-power wireless communications," *IEEE Transactions on Circuits and Systems I*, vol. 51, no. 2, pp. 371-378, February 2004.

[2] A. Viterbi, "Error bounds for convolutional codes and asymptotically optimum decoding algorithm," *IEEE Transactions on Information theory*, vol. It-13, no. 2, pp. 260–269, April 1967.

[3] G. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.

[4] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1995.

[5] H.-D. Lin and D. Messerschmitt, "Algorithms and architectures for concurrent viterbi decoding," *Proc. IEEE International Conference on Communications*, pp. 836-840, June 1989.

[6] K. Page and P. Chau, "Improved architectures for the add-compare-select operation in long constraint length viterbi decoding," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 1, pp. 151-155, January 1998.

[7] C.-Y. Tsui, R.-K. Cheng, and C. Ling, "Low power acs unit design for the viterbi decoder," *Proc. IEEE International Symposium on Circuits and Systems*, pp. 137-140, 1999.

[8] J.-S. Han, T.-J. Kim, and C. Lee, "High performance viterbi decoder using modified register exchange methods," *Proc., IEEE International Symposium on Circuits and Systems*, pp. III 553-556, May 2004.

[9] J. H. et al, "Cdma mobile station modem asic," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 3, pp. 253-260, March 1993.

[10] I. Kang and A. W. Jr, "Low-power viterbi decoder for cdma mobile terminals," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 473-482, March 1998.

[11] Y.-N. Chang, H. Suzuki, and K. Parhi, "A 2-mb/s 256-state 10-mw rate-1/3 viterbi decoder," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 826-834, June 2000.

[12] R. Henning and C. Chakrabarti, "Low-power approach for decoding convolutional codes with adaptive viterbi algorithm approximations," *Proc. IEEE International Symposium on Lower Power Electronics and Design*, pp. 68-71, August 2002.

[13] J. B. Anderson and S. Mohan, *Source and Channel Coding: An Algorithmic Approach*. Kluwer Academic Publishers, 1991.

[14] S.-S. Wang, "A state-reduction viterbi decoder for convolutional codes with large constraint lengths," Master's thesis, National Chiao Tung University, Hsinchu, Taiwan, June, 2002.

[15] H.-L. Lou, "Linear distances as branch metrics for viterbi decoding of trellis codes," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3267-3270, June 2000.

[16] J. Ryu, S. Kim, J. Cho, H. Park, and Y. Chang, "Lower power viterbi decoder architecture with a new clock-gating trace-back unit," *Proc. 6th International Conference on VLSI and CAD*, pp. 297-300, Oct. 1999.

Dalia El-Dib received her B.Sc. and M.Sc. degrees in Electrical Engineering from Cairo University, Cairo, Egypt, in 1995 and 1998, respectively. From 1996 to 1999, she worked for the Electronic Research Institute, and was a part-time Software Engineer for the Information and Decision Support Center, Cairo, Egypt. Her M.Sc. research focused on reconfigurable microprocessor architectures. Then, she attained her doctoral degree from University of Waterloo, Ontario, Canada in 2004. Now, she is with the Electronics Research Institute, Cairo, Egypt. Her research interests include among others low-power design techniques at the algorithm/architecture level, especially for coding/decoding circuits in wireless communications.

M.I. Elmasry (F'88) was born in Cairo, Egypt on December 24, 1943. He received the B.Sc. degree from Cairo University, Cairo, Egypt, and the M.A.Sc. and Ph.D. degrees from the University of Ottawa, Ottawa, Ontario, Canada, all in Electrical Engineering, in 1965, 1970 and 1974 respectively. He has worked in the area of digital integrated circuits and system design for the last 35 years. He was with Cairo University from 1965 to 1968, and Bell-Northern Research, Ottawa, Canada from 1972 to 1974. He has been with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ont., Canada, since 1974, where he is a Professor and founding Director of the VLSI Research Group. He has held the NSERC/BNR Research Chair in VLSI design at the same university from 1986 to 1991. He has served as a consultant to research laboratories in Canada, Japan and the United States. He has authored and co-authored over 400 papers and 14 books on integrated circuit design and design automation. He has several patents to his credit. He is founding President of Pico Electronics Inc. Dr. Elmasry has served in many professional organizations in different positions and received many Canadian and International Awards. He is a founding member of the Canadian Conference on VLSI, the Canadian Microelectronics Corporation (CMC), the International Conference on Microelectronics (ICM), MICRONET and CITO. He is a Fellow of the IEEE, the Royal Society of Canada and of the Canadian Academy of Engineers.