ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2014

Low Power Error Control Coding Implementation for Wireless Sensor Networks

Dr.T.Saravanan

HOD, Department of ETC, Bharath University,Chennai-600073, India

ABSTRACT- Wireless sensor networks (WSN) offer an increasingly attractive mode of data gathering in distributed system architectures and dynamic access via wireless connectivity. ECC provides coding gain, resulting in transmitter energy savings, at the cost of added decoder power consumption. The main challenge in deploying WSN is to improve energy-efficiency and lifetime of the nodes while keeping communication reliability. The energy efficiency of error control schemes should be considered because of the strict energy constraints of wireless sensor networks. Wireless sensor networks require simple and facile error control schemes because of the low complexity request of sensor nodes. With the ever increasing data throughputs required by communication application, there is an actual need for new effective architectures (small area and high speed) for circuit parts dedicated to error detecting/correcting coding (EDC/ECC).The Convolutional Encoder and Decoder for Wireless Sensor Networks are studied and the right configurations for the encoder will be selected and a Parallel/Pipelined Architecture for the Convolutional Encoder will be explored and implemented in an FPGA platform in order to provide a low power and fast encoding scheme and the best architecture will presented.

I. INTRODUCTION

In general, WSN nodes are made of battery-supplied small devices with reduced processing capability and a radio frequency transceiver unity, both operating in a collaborative way [1]. Therefore, much of the research in this area is concerned with energy conservation. The goal is to extend the
Sensor node and network lifetimes, since the lost of a node can make the network unavailable. However, the aforementioned solutions are susceptible to channel impairments, because any radio signal is affected by random noise and channel fading [2, 3]. If a node receives a corrupted data packet, the data can be discarded and the node keeps waiting for a new transmission or the node employs an Automatic Repeat request (ARQ) procedure (a retransmission procedure). However, in both cases there is a waste of energy in the network. A particularly undesirable situation occurs when the channel condition is bad, causing successive retransmissions.

Another method to increase the energy conservation in WSN is to apply forward error correction (FEC) strategies, reducing the frame error rate and consequently the number of retransmissions. Basically there are two classes of error control codes: block codes and convolutional codes. The convolutional encoding technique is a strategy widely used in wireless communication environments like sensor networks, since they usually present a simpler implementation for the same performance of a competitor block code [4]. The convolutional encoder is implemented using a set of shift registers (memories) and module two adders. The efficiency of a convolutional code depends on its memory order and coding rate. Convolutional codes can be decoded by using the code trellis to find the most likely sequence of codes. The Viterbi Algorithm (VA) [4] simplifies the decoding task by limiting the number of sequences examined. In a preliminary analysis it seems to be adequate to apply a powerful error correcting code in all network nodes in a WSN, in order to obtain the maximum error correction capability. In our study, we considered FEC schemes employing convolutional codes with rate 1/2 for different complexities (memory orders). The ARQ scheme is assumed to follow a stop-and-wait protocol [6]. The main objective is to demonstrate the improvement in the network energy conservation through the use of optimized code rate selection.

In this paper, we introduce a new architectural scheme for the OTM convolutional encoders, in which parallel and pipelining techniques are used together. While these approaches are generally successful in matching the high throughput constraints, they generally tend to miss the low-cost constraints of end user applications.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2014

II. RADIO CHANNEL MODEL

In a communication process, the transmitted signal suffers path loss attenuation and is corrupted by Additive White Gaussian Noise (AWGN). In a wireless environment there is an additional degradation generated by multipath fading, causing fluctuations in the received signal strength. Multipath
Fading significantly decreases WSN performance in terms of energy consumption, because in general the packets received with errors must be retransmitted. The fading process is classically modeled using the Rayleigh probability distribution [2]. The performance of a sensor node in terms of frame or bit error probability depends on the average Signal-to-Noise Ratio (SNR) at the receiver. By its turn the instantaneous SNR value depends on the channel gain during a symbol or block transmission, which follows the Rayleigh distribution. The path loss model considered in this work is defined
by [2]

$$PL(d)\,[dB] = PL(d_0)\,[dB] + n\,10\log_{10}(d/d_0) + \chi\sigma \quad (1)$$

Where, *PL (d)* represents the path loss at a distance *d* from the transmitter. The parameter *PL (do)* defines the path loss in a reference distance *do* and *n* is the environment path loss exponent, usually between 2 and 4 [2]. The parameter $\chi\sigma$ is a random variable with log-normal distribution, which represents the shadowing effect in the received signal, i.e., fluctuations in the path loss value for the same distance between transmitter and receiver. The received average signal power, *Prx*, as a function of the distance *d* between transmitter and receiver is given by

$$Prx\,[dBm] = Ptx\,[dBm] - PL(d)\,[dB], \quad (2)$$

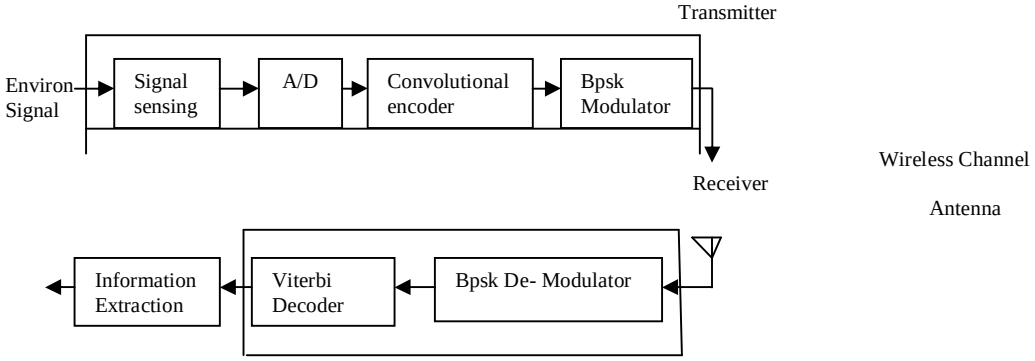
Where, *Ptx* is the transmitted power.

At the receiver, the SNR is defined as $\gamma = Prx/Pnoise$, Where $Pnoise = No.\ B$ is the noise power, *No* is the unilateral Noise power spectral density (W/Hz), and *B* is the receiver bandwidth (Hz). This bandwidth depends on constructive characteristics of the receiver. The received power, *Prx*, may be expressed as $Prx = Eb.\ Rb$, where *Eb* is the average coded bit energy and *Rb* is the raw transmission rate. Therefore the received SNR can also be specified as

$$\gamma = Eb \cdot Rb / No \cdot B. \quad (3)$$

A sensor node generally employs a low cost and low complexity radio transceiver. In our investigation, we consider Binary Phase Shift Keying (BPSK) modulation. The typical low transmission rate of a sensor node leads to a slowly-varying fading channel model [7]. The duration *b*size of a block-fading period is determined by the channel coherence time. The number of periods can be controlled through a slow frequency hopping systems at the transmitter. Our analysis of the bit error rate (BER) for the block fading channel was carried out through computer simulations. Then, the frame error rate (FER) can be estimated for a specific data frame size as $FER = 1-(1-BER)f^{d}_{size}$,where f^{d}_{size} denotes the data frame size in bits. The coded data frame size is defined as $f^{c}_{size} = f^{d}_{size}/r$, where *r* is the coding rate.

III. SYSTEM FLOW MODEL



ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2014

A) Signal Sensing:

A sensor node is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. The main components of a sensor node are a microcontroller, transceiver, power source .

B) Convolutional Encoder:

Convolutional codes are frequently used to correct errors in noisy channels. They have rather good correcting capability and perform well even on very bad channels (with error probabilities of about 10−3). Convolutional codes are extensively used in satellite communications.

i) Encoder Structure:

A convolutional code introduces redundant bits into the data stream through the use of linear shift register. A convolutional encoder [8] is specified by the parameters *(n, k,m, d*free*),* where *n* is the number of output bits, *k* is the number of input bits, *m* is the memory order and *d*free is the free distance of the code, which is defined as the minimum Hamming distance between two coded sequences. The error correction capability is a function of the free distance. The information bits are input into shift registers and the output encoded bits are obtained by modulo-2 addition of the input information bits and the contents of the shift registers. The connections to the modulo-2 adders were developed heuristically with no algebraic or combinatorial foundation.

The code rate r for a convolutional code is defined as r=k/n

Where k is the number of parallel input information bits and n is the number of parallel Output encoded bits at one time interval. The constraint length K for a convolutional code is defined as K = m+ 1, m is the maximum number of stages (memory size) in any shift register. Convolutional code can become very complicated with various code rates and constraint lengths.

The encoder can be represented in several different but equivalent ways.

1) Generator Representation:

Generator representation shows the hardware connection of the shift register taps to the modulo-2 adders. A generator vector represents the position of the taps for an output. A “1” represents a connection and a “0” represents no connection.

2) Tree Diagram Representation:

The tree diagram representation shows all possible information and encoded sequences for the convolutional encoder. In the tree diagram, a solid line represents input information bit 0 and a dashed line represents input information bit 1.

3) State Diagram Representation:

The state diagram shows the state information of a convolutional encoder. The state information of a convolutional encoder is stored in the shift registers. In the state diagram, the state information of the encoder is shown in the circles. Each new input information bit causes a transition from one state to another. The path information between the states, denoted as x/**c**, represents input information bit x and output encoded bits **c**.

4) Trellis Diagram Representation:

A convolutional encoder is often seen as a finite state machine. Each state corresponds to some value of the encoder's register. Given the input bit value, from a certain state the encoder can move to two other states. These state transitions constitute a diagram which is called a trellis diagram.

C) BPSK Modulator:

In BPSK, individual data bits are used to control the phase of the carrier. During each bit interval, the modulator shifts the carrier to one of two possible phases, which are 180 degrees or π radians apart. This can be accomplished very simply by using a bipolar baseband signal to modulate the carrier’s amplitude. The output of such a modulator can be represented mathematically as x (t) = R (t) cos (ωct + θ), Where R (t) is the bipolar baseband signal, ωc is the carrier frequency, and θ is the Phase of the unmodulated carrier.

D) BPSK Demodulator:

The modulated signal is multiplied by the recovered carrier, and this product is integrated over a bit interval. If the integration result is positive, the received bit is deemed to be 1; if the integration result is negative, the received bit is deemed to be 0. The recovered carrier input to the model is in the form of a real-valued sinusoid, and the recovered clock input to the model is in the form of an integer-valued sequence that has zero values everywhere at the sampling instants corresponding to the end of each bit interval.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2014

E) Viterbi Decoder:

The Viterbi algorithm is based on the principle of maximum likelihood decoding, which in the present case is equivalent to minimum distance decoding. Upon reception of a sequence of bits, the particular path through this diagram will be searched which is closest to this sequence in the sense of Hamming distance. Convolutional encoding is a simple procedure, decoding of a convolutional code is much more complex task. Viterbi decoding is an optimal (in a maximum-likelihood sense) algorithm for decoding of a Convolutional code. Its main drawback is that the decoding complexity grows exponentially with the code length. So, it can be utilized only for relatively short codes. A soft decision decoder is a decoder receiving bits from the channel with some kind of reliability estimate. Three bits are usually sufficient for this task.

A hard decision decoder – a decoder which receives only bits from the channel (without any reliability estimate).

IV. THEORETICAL DECODING COMPLEXITY

A convolutional code can be represented by a trellis diagram. This describes the permissible encoder states and transitions. The most employed convolutional decoding method is the Viterbi algorithm, which operates over the code trellis. As the transition between two states $(Si,Si+1)$ defines the encoder output, the Viterbi algorithm evaluates at each instant the most likely next state, according to a previous metric of each initial state Si , and the cost of each encoder state transition which is calculated based on the received data. The computational effort of the Viterbi algorithm is proportional to the density of the trellis module, or the trellis complexity. The trellis module has 2^m states. Each initial state is connected to 2^k final states. Therefore, there are a total of 2^{m+k} branches in a trellis module. Since each branch is labeled by n bits, in a trellis module there are a total of $n \cdot 2^{m+k}$ symbols [9].

$$C = n/k \cdot 2^{m+k} \text{ symbols/bit.} \qquad (4)$$

Figure.1shows the decoding complexity C for three different Coding rates as a function of the code memory. The values are normalized by the decoding complexity of a convolutional code with parameters $(n, k, m) = (2, 1, 1)$. This complexity is proportional to the number of instructions (processing energy consumption) that must be executed by the receiver processing unit in order to decode a data frame. In our investigation we considered convolutional codes with rate $r = 1/2$ and different memory orders.

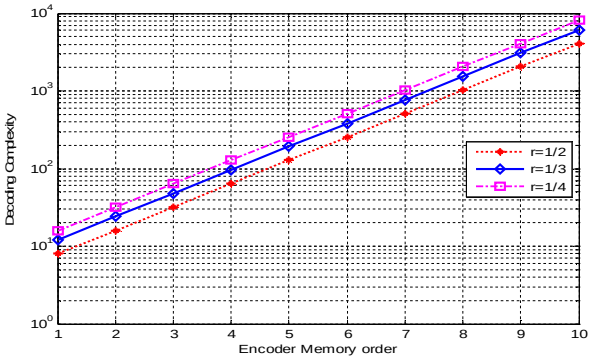


Fig. 1-Normalized theoretical decoding complexity of several convolutional codes

Figure.2 shows a comparison between the theoretical decoding

Complexity metric (symbols/bit) given by $C = n/k \cdot 2^{m+k}$ symbols/bit (1). And the practical decoding complexity (instructions cycles/bit) given by

$$IC_{total} = (5 \cdot 2^{m-1} + 2^{m-4} + n \cdot 2^{n-1} + 16.25) \text{ IC/bit} \qquad (5)$$

It was assumed, without loss of generality, that the decoding of one symbol requires one instruction and the practical decoding complexity cycle (IC). In practice, the number of instruction cycles required per symbol depends on the specific platform implementation.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2014

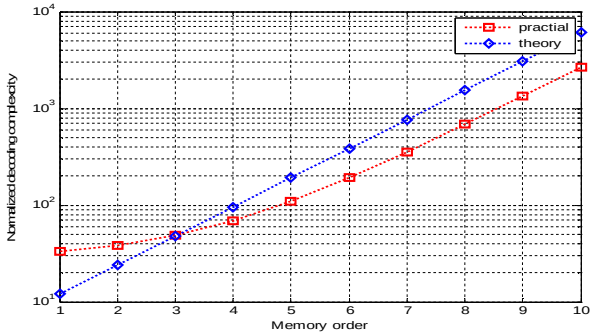


Fig. 2-Comparison between the theoretical decoding complexities Given by (2) and the practical decoding complexity given by (2)

V. CONVOLUTIONAL CODING PERFORMANCES

Increasing the code complexity also increases the decoding Energy consumption (disadvantage) but results in a better error correction capability (advantage). In this sense, our study aims at determining the optimal choice for the convolutional code complexity used in the communication between two sensor nodes, in order to achieve the best trade-off between energy consumption and error correcting capability. We derived, through computer simulations, the performance of a rate r = 1/2 convolutional code with different memory orders for the block fading Rayleigh channel as defined in Sect. 2. Figure 3 shows the performance curves in terms of frame error rate (FER). From the figure it is clear that the performance improves with the memory order.

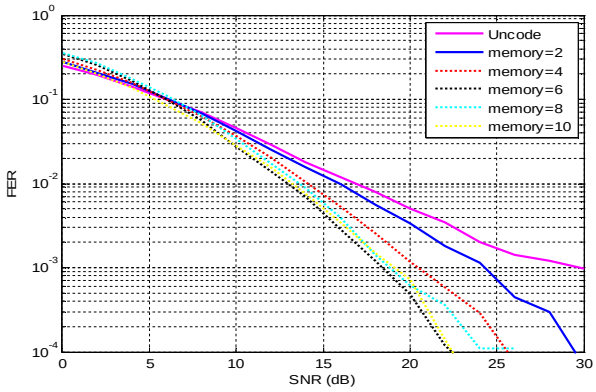


Fig. 3-Performance in terms of FER versus SNR for several rates 1/2 Convolutional codes of different memory orders *m*

VI. SERIAL ARCHITECTURE

The serial form of a OTM convolutional encoder is shown in Fig. 4 for m = 3, where m is the encoder memory size. Inputs and outputs at time t are respectively equal to (X)t and (Y)t. The notational ()t is used here to denote the content of the referenced line at the beginning of clock cycle t. It will be used henceforth throughout the paper.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2014

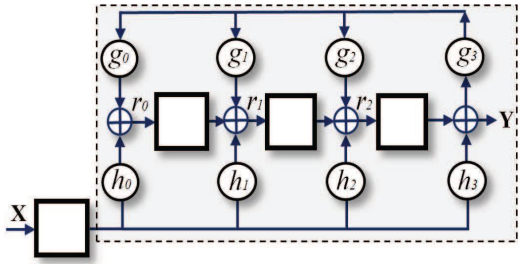


Fig 4-OTM serial encoder for m = 3.

Two types of OTM encoder exist: the Non Recursive Convolutional (NRC) and the Recursive Convolutional (RC). The encoder type (NRC or RC) is set according to the following equation [1]:

$$g_3 = 1 \Rightarrow \begin{cases} (g_0, g_1, g_2) = (0,0,0) \Rightarrow NRC \\ (g_0, g_1, g_2) \neq (0,0,0) \Rightarrow RC \end{cases} \qquad (6)$$

Recursive convolutional (RC) codes differ from non-recursive Convolutional (NRC) codes by the fact that the values in R = (r0, r1, r2) are not only driven by the input X, through the feed forward generator H = (h0, h1, h2, h3), but also by the output Y through a feedback loop controlled by the feedback generator G = (g0, g1, g2, g3).

VII. PARALLEL-PIPELINED ARCHITECTURE

The architecture presented hereafter, whose overall scheme is shown in Fig. 5, is a fast small-area parallel pipeline encoder to be used with convolutional codes. In the following, we will describe its operation principles and its application to OTM convolutional encoders. The CPS (Cumulated Pipeline State) acronym will be used down from here to designate its constitutive block, whose role is explained next.

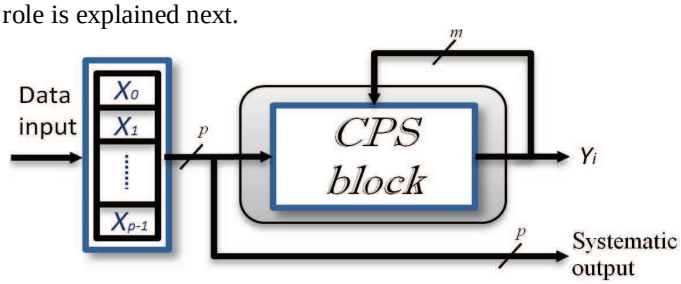


Fig.5- parallel-pipelined OTM encoder.

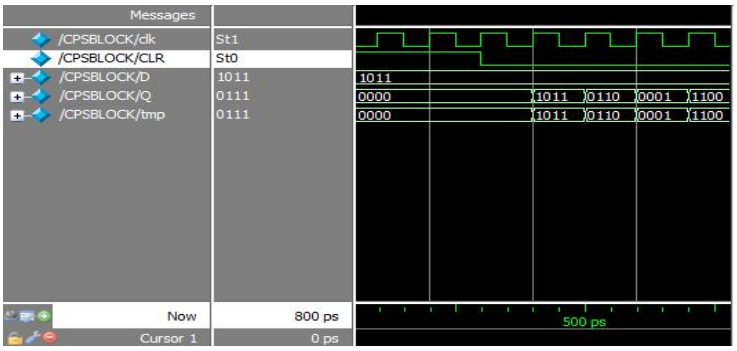


Fig.6-CPS Block

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2014

A parallel-pipelined approach is used to process p bits per clock cycle (p > 1), while preventing a slowdown of the clock frequency f_{clk}. As depicted in Fig. 5, the architecture is build up of just the CPS block. The structure of this block generates the p output bits of the OTM encoder combining delayed values (in the pipeline) of the data on the inputs X_i and the outputs Y_i using multiple feed-forward and feedback loops built over the coefficients of the H and the G generators, respectively.

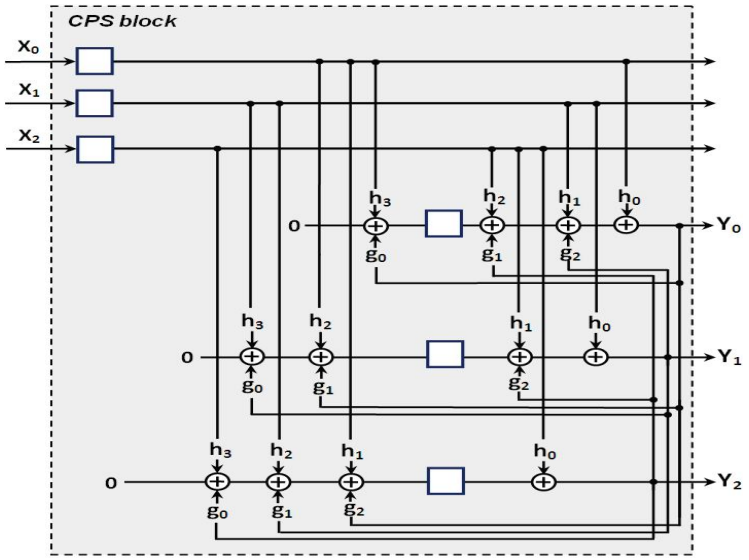


Figure 7: Parallel-pipeline OTM architecture for m = 3

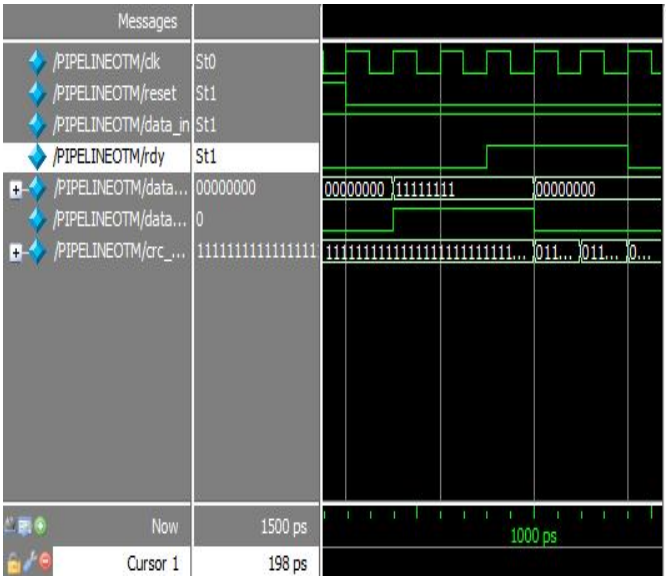


Figure 8: Parallel-pipeline OTM architecture

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2014

VIII. CONCLUSION

The use of convolutional codes for error correction in wireless sensor networks can significantly increase the node and network lifetime, if the optimized code complexity is employed along each transmission hop. The use of the optimized complexity reduces the overall energy consumption generated by the local node processing and by the HARQ retransmission protocol. It is important to say that the savings in energy consumption depend on the network topology.

In this paper, we proposed new parallel-pipeline architecture for OTM (One To Many) convolutional encoders. Actually, in addition to global speed (throughput) acceleration, meaningful area savings were made possible. On 32-bit parallel-pipeline implementations, up to about 58% area savings have been achieved with data rates up to 8.10 Gbits/s. Practically, the new architectural approach proves to be an Effective method in general to design small-area and high throughput convolutional encoders, able to satisfy the low cost and high-speed constraints characterizing modern digital Communication systems.

REFERENCES

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks Journal*, *38*(4), 393–422.
[2] Rappaport, T. S. (1996). *Wireless communications: principles and Practice.* New York: Prentice Hall.
[3] Proakis, J. G. (1995). *Digital communications*. New York: McGraw-Hill.
[4] Hendrix, H. (2002). Viterbi decoding techniques for the TMS320C54x DSP generation. In *Texas instruments application report SPRA071A*.
[5] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *13*, 260–269.
[6] Wicker, S. B. (1995). *Error control systems for digital communication and storage*. New York: Prentice Hall.
[7] Chuang, A., Guillen, A. F., Rasmussen, L. K., & Collings, I. B. (2006). Optimal rate-diversity-delay tradeoff in ARQ block-fading channels. In *Proceedings of the IEEE information theory workshop*, Uruguay (pp. 507–511).
[8] Lin, S., & Costello, D. J. (1983). *Error control coding: fundamentals and applications*. New York: Prentice Hall.
[9] McEliece, R. J., & Lin, W. (1996). The trellis complexity of Convolutional codes. *IEEE Transactions on Information Theory*, *42*(6), 1855–1864.
[10] Hendrix, H. (2002). Viterbi decoding techniques for the TMS320C54x DSP generation. In *Texas instruments application report SPRA071A*.
[11] A. M'sir, F. Monteiro, A. Dandache, B. Lepley, "Design of a high speed parallel encoder for convolutional codes," Microelectronics Journal, vol. 35(2), pp. 151–166, Feb. 2004.
[12] Karvonen, H., Shelby, Z., & Pomalaza-Raez, C. (2004). Coding for energy efficient wireless embedded networks. In *Proceedings of the IEEE international workshop on wireless ad-hoc networks*, Finland (pp. 300–304).
[13] Keshab K.Parhi.(1999), "VLSI Digital Signal Processing Systems" Design and Implementation, by John Wiley & Sons.