

LOW-POWER CHANNEL CODING VIA DYNAMIC RECONFIGURATION

Manish Goel and Naresh R. Shanbhag

Coordinated Science Lab/ECE Department,
Univ. of Illinois at Urbana-Champaign,
1308 W. Main Street, Urbana IL-61801.
E-mail : [mgoel,shanbhag]@uivlsi.csl.uiuc.edu

ABSTRACT

Presented in this paper are *energy-optimum reconfiguration strategies* for channel codecs. These strategies are derived by solving an optimization problem, which has energy consumption as the objective function and a constraint on the bit error-rate (*BER*). Energy consumption models for a reconfigurable Reed-Solomon (RS) codec are derived via gate-level simulation of the finite field arithmetic modules. These energy models along with the *BER* expressions are then employed to derive the energy-optimum reconfiguration strategies. The energy savings are computed by comparing the energy consumption of the reconfigurable codec with that of the static codec. The energy savings range from 0%-83% for channel signal-to-noise ratio (*SNR*) variations from 7dB-10dB. On an average 55% energy savings are achieved.

1. INTRODUCTION

Power-reduction techniques have been proposed at all levels of VLSI design hierarchy ranging from the circuits to algorithms. Of particular interest in this paper are *algorithm transformation techniques* [1].

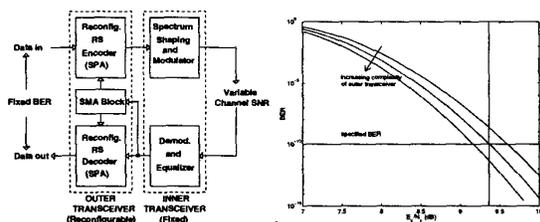


Figure 1: (a) DAT-based communication system and (b) *BER* versus E_b/N_0 .

While *static algorithm transforms* (SAT) [2] are applied during the algorithm design phase and assuming a worst-case stationary input, we have proposed *dynamic algorithm transforms* (DAT) [3] that optimize the energy consumption of a DSP system via run-time reconfiguration by exploiting input non-stationarities. In this paper, we present DAT-based channel coding which achieve energy savings in the presence of variable channel *SNR*.

Consider a digital communication system shown in Fig. 1(a). The *inner transceiver* in Fig. 1(a) employs spectrum shaping (at transmitter) and equalization (at receiver) to achieve a specified *SNR* at

its output, denoted by E_b/N_0 , where E_b is energy per bit and N_0 is single-sided power spectral density of noise. The *outer transceiver* employs error-correction to achieve a specified *BER*. Thus, in Fig. 1(b), the position of the vertical line and horizontal lines are fixed by the inner and outer transceivers, respectively. The goal of communication system design is to achieve a specified end-to-end *BER* (which is fixed by the application). Conventional practice is to design for the *worst-case channel SNR*. This design is usually complex and is an "over-kill" (in terms of *BER*) for the *best* and *nominal* cases. This is because an improvement in channel *SNR* (shorter distances) gives better *BER*. This slack in *BER* performance can be traded off by reconfiguring the transceiver, and thus achieving energy savings. This can be done in two ways:

(1) by reconfiguring the inner transceiver such that a constant E_b/N_0 and therefore, a constant *BER* can be achieved by a fixed outer transceiver (i.e. by operating on a fixed *BER* curve in Fig. 1(b)), or

(2) by keeping the inner transceiver fixed (thus obtaining a variable E_b/N_0) and by reconfiguring the outer transceiver (i.e. by going from one performance curve to the other in Fig. 1(b)), a constant *BER* can be achieved.

In past, we studied scenario (1) and have shown that the energy savings ranging from 60%-90% can be achieved via a DAT-based reconfiguration of the equalizer in an ATM-LAN [4] and very high-speed digital subscriber loop (VDSL) [3] environments. In the current work, we focus on scenario (2) above where, we assume that the inner transceiver is fixed, and the outer transceiver is a reconfigurable Reed-Solomon (RS) codec.

Low-power RS codecs employing *look-ahead* [5] and *digit-serial* [6] architectures have been presented. In addition, the programmable codecs based on ASIC [7, 8] and FPGA [9] implementation style have been presented in the literature. In this paper, we employ the DAT approach for reconfiguring RS codecs. The unique features of the DAT approach are:

(1) it is hardware-platform independent because platform-specific details (such as architecture and energy consumption) are made part of the reconfiguration strategy, and

(2) the hardware energy consumption is optimized under system level constraints (such as *BER*).

In the next section, we apply the DAT framework [3] to channel coding. In section 3, we present the energy consumption models for the RS codec. The reconfiguration strategy and the energy savings are presented in section 4.

This work was supported by DARPA contract DABT63-97-C-0025 and NSF CAREER award MIP-9623737.

2. DAT FRAMEWORK FOR CHANNEL CODING

In this section, we present DAT framework for channel coding. A DAT-based VLSI signal processing system (see Fig. 1) has two subsystems: 1.) a signal processing algorithm (SPA) block that is a reconfigurable datapath; and 2.) a signal monitoring algorithm (SMA) block that implements a reconfiguration strategy/control, based upon temporal/spatial variabilities in the input. The input variabilities naturally lead to the definition of input states and the underlying probability distribution of these states, defined as follows:

Definition 1 : The input state $\mathbf{s}(n) \in \mathcal{S} = \{s_1, s_2, \dots, s_{N_s}\}$ (where \mathcal{S} is the state-space), at time instant n is a vector of input-dependent parameters where $\mathbf{s}(n) = s_i$ with a probability $p(s_i)$.

For example, for channel coding, the input state can be defined as follows:

$$\mathbf{s}(n) = [E_b/N_o], \quad (2.1)$$

where E_b/N_o is the signal-to-noise ratio at the output of the inner transceiver. Assume that E_b/N_o varies from 7dB to 10dB. The state space \mathcal{S} can be defined as:

$$\mathcal{S} = [s_1, s_2, \dots, s_{16}] = [7dB, 7.2dB, \dots, 10dB], \quad (2.2)$$

where $N_s = 16$ states are defined.

Definition 2 : The configuration $\mathbf{c}(n) \in \mathcal{C} = \{c_1, c_2, \dots, c_{N_c}\}$ (where \mathcal{C} is the configuration space) at time instant n is defined as a vector of reconfiguration control signals. Each configuration vector \mathbf{c}_i corresponds to a particular value of the control signals.

In case of the reconfigurable RS encoder in Fig. 2, the configuration vector $\mathbf{c}_{enc}(n)$ can be defined as:

$$\mathbf{c}_{enc}(n) = [\alpha_{enc,0}, \alpha_{enc,1}, \dots, \alpha_{enc,2T-1}], \quad (2.3)$$

where $\alpha_{enc,i} \in \{0, 1\}$ are the configuration/control signals, and T is the maximum error correction capability of the encoder. The control signal values $\alpha_{enc,i} = 0$ and $\alpha_{enc,i} = 1$ powers-down and powers-up the i^{th} tap in the encoder, respectively. Thus, the configuration vector size depends upon the actual hardware architecture. If $t(n)$ is the error-correction capability of the encoder at time instant n , then the configuration $\mathbf{c}_{enc}(n)$ is related to $t(n)$ as follows:

$$\sum_{i=0}^{2T-1} \alpha_{enc,i} = 2t(n), \quad (2.4)$$

which is also equal to the number of powered-up taps. Thus, the

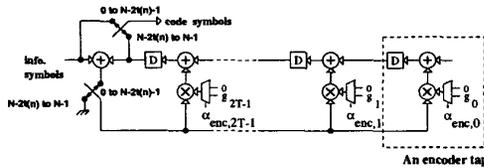


Figure 2: A reconfigurable RS encoder.

energy savings can be achieved by reducing t , and thus, number of powered-up taps in the encoder. Similarly, the configuration signals for the RS decoder (Fig. 3) can be defined. Without going into the details of the decoder architecture, we mention that the configuration vector for the decoder is defined as,

$$\mathbf{c}_{dec}(n) = [\alpha_{sc,0}, \alpha_{sc,1}, \dots, \alpha_{sc,2T-1}, \alpha_{bm,0}, \alpha_{bm,1}, \dots, \alpha_{sc,T-1}, \alpha_{cf,0}, \alpha_{cf,1}, \dots, \alpha_{cf,T-1}], \quad (2.5)$$

where $\alpha_{sc,i}, \alpha_{bm,i}$ and $\alpha_{cf,i}$ are the control signals to power-up/down the i^{th} sub-block in syndrome computation (SC) block, Berlekamp-Massey (BM) block and Chien-Forney (CF) block, respectively. The control signal $\alpha_{X,i} = 0$ and $\alpha_{X,i} = 1$ implies that i^{th} sub-block in block X is power-up and power-down, respectively. If $t(n)$ is the error correction capability required at time n , then $\mathbf{c}_{dec}(n)$ is related to $t(n)$ as follows:

$$\frac{1}{2} \sum_{i=0}^{2T-1} \alpha_{sc,i} = \sum_{i=0}^{T-1} \alpha_{bm,i} = \sum_{i=0}^{T-1} \alpha_{cf,i} = t(n). \quad (2.6)$$

The codec configuration signal can be defined by combining $\mathbf{c}_{enc}(n)$

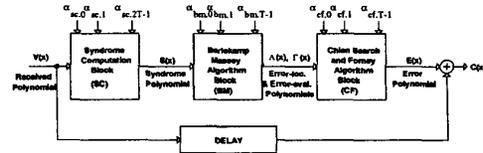


Figure 3: A reconfigurable RS decoder.

and $\mathbf{c}_{dec}(n)$ in (2.3) and (2.5) as follows:

$$\mathbf{c}_{codec}(n) = [\mathbf{c}_{enc}(n), \mathbf{c}_{dec}(n)]. \quad (2.7)$$

It can be shown that for reconfigurable codec with configuration vector $\mathbf{c}_{codec}(n)$ given by (2.7), the number of possible configurations N_c equals 2^{6T} . Thus, the number of possible configurations explodes exponentially with increase in number of configuration parameters. Therefore, we need a systematic method for finding the energy-optimum configuration vector $\mathbf{c}_{opt}(s_i)$ defined as follows:

Definition 3 : The energy-optimum configuration $\mathbf{c}_{opt}(s_i) \in \mathcal{C}$ for a given input state $s_i \in \mathcal{S}$ is defined as:

$$\mathbf{c}_{opt}(s_i) = \arg \min_{\mathbf{c} \in \mathcal{C}} \mathcal{E}_{SPA}(\mathbf{c}), \quad (2.8)$$

s.t. $\mathcal{J}_{SPA}(s_i, \mathbf{c}) \leq \mathcal{J}_o$,

where $\mathcal{E}_{SPA}(\mathbf{c})$ is energy dissipated by the SPA block in configuration \mathbf{c} , \mathcal{J}_o is the specified BER and $\mathcal{J}_{SPA}(s_i, \mathbf{c})$ is the BER achieved by the SPA block when the input is in state s_i and the SPA block is in configuration \mathbf{c} .

Note that the objective function in (2.8) is a VLSI parameter (energy) and the constraint is a system parameter (BER). For example, if the SPA block in the above definition is a channel codec, one can specify the desired BER $\mathcal{J}_o = 10^{-10}$. The bit error rate $\mathcal{J}_{SPA}(s_i, \mathbf{c})$ can be computed. Assume that the inner transceiver in Fig. 1 is fixed and employs binary phase-shift keying (BPSK) modulation scheme. The raw bit-error probability for this scheme is given by, $\epsilon = Q[\sqrt{E_b/N_o}]$, where $Q[x]$ is the probability that a standard Gaussian random variable lies between x and ∞ . The symbol error probability is given by,

$$p_s = 1 - (1 - \epsilon)^m = 1 - \left(1 - Q\left[\sqrt{E_b/N_o}\right]\right)^m. \quad (2.9)$$

If $t(n)$ is the error correction capability of RS codec (SPA block) at time instant n , then the bit error probability $\mathcal{J}_{SPA}(s_i, \mathbf{c})$ at time

instant n is given by,

$$\mathcal{J}_{SPA}(s_t, c) = \sum_{j=t(n)+1}^N \binom{N}{j} p_s^j (1-p_s)^{N-j}, \quad (2.10)$$

where N is the length (in symbols) of each codeword and we have summed the probabilities of all terms corresponding to more than $t(n)$ symbol errors occur. This is valid under the assumption that the received codeword is completely discarded whenever there is a decoding error. The term p_s in (2.10) depends upon the state $s(n)$ via (2.1) and (2.9). The term $t(n)$ is related to the SPA configuration by (2.4) and (2.6). In the next section, we evaluate the energy consumption of the SPA block in configuration $c(n)$ i.e. $\mathcal{E}_{SPA}(c)$.

3. ENERGY CONSUMPTION MODELS FOR RECONFIGURABLE RS CODEC

In this section, we find the energy consumption models for the Galois Field (GF) arithmetic modules, RS encoder and decoder. These energy models are then employed in (2.8) to derive the reconfiguration strategy (in section 4).

3.1. GF Arithmetic Modules

Each element in $GF(2^m)$ is defined by a m -bit binary vector. The main GF operations are addition, multiplication and inversion. For details on these operations, the interested readers are referred to [10].

The addition over $GF(2^m)$ is defined as bit-wise sum of the m bits of two operands requiring m XOR gates. It is assumed that the standard cells based $0.18\mu m$, $2.5V$ CMOS technology are being employed. The adder circuit was simulated via gate-level simulator MED [11]. It was found that the energy consumption per m -bit addition is given by,

$$\mathcal{E}_{add} = 3.3 \times 10^{-5} m \text{ (mW/MHz)}. \quad (3.1)$$

For multiplication, a bit-parallel architecture given in [6] was assumed. This architecture requires approximately $2m^2$ XOR gates and $2m^2$ AND gates for each multiplier. It was found by MED simulation tool [11] that the energy consumption of the $m \times m$ -bit multiplier is given by,

$$\mathcal{E}_{mult} = 3.7 \times 10^{-5} m^3 \text{ (mW/MHz)}. \quad (3.2)$$

If a is a $GF(2^m)$ element, then its inverse is computed as follows:

$$a^{-1} = a^{2^m-2} = a^2 \otimes (a^2)^2 \otimes \left((a^2)^2 \right)^2 \otimes \dots (m-1) \text{ times}, \quad (3.3)$$

which involves $(2m-3)$ multiplications. Therefore, the energy consumption per m -bit inversion is given by,

$$\mathcal{E}_{inv} = 3.7 \times 10^{-5} (2m-3)m^3 \text{ (mW/MHz)}. \quad (3.4)$$

In the following subsections, we employ the energy consumption expressions in (3.1)-(3.4) to derive the expression for the energy consumption of an RS encoder and decoder.

3.2. RS Encoder

A reconfigurable RS encoder based on the parallel architecture is shown in Fig. 2. We will compute the energy consumption of this encoder at time n , when the configuration $c_{enc}(n)$ is given by (2.3). It can be shown that in each of the first $N-2t(n)$ clock cycles, $\sum \alpha_{enc,i}$ multiply-adds (same as number of powered up taps) are carried out. Therefore, the energy consumed per codeword generation is given by:

$$\mathcal{E}_{enc} = (N-2t(n))(\mathcal{E}_{mult} + \mathcal{E}_{add}) \sum_{i=0}^{2T-1} \alpha_{enc,i}, \quad (3.5)$$

where \mathcal{E}_{add} and \mathcal{E}_{mult} are given by (3.1) and (3.2), respectively. This can be simplified via (2.4) as,

$$\mathcal{E}_{enc} = 2t(N-2t)(\mathcal{E}_{mult} + \mathcal{E}_{add}), \quad (3.6)$$

where we have removed the time variable n for simplicity of notation.

3.3. RS Decoder

The block diagram of RS decoder is shown in Fig. 3. The syndrome computation (SC) block computes $2t(n)$ syndromes from the received codeword. These syndromes are then passed to the block implementing Berlekamp-Massey (BM) algorithm, which generates the error-locator and the error-evaluator polynomials. These polynomials are then processed by the last block, which implements Chien's search algorithm for detecting error locations, and Forney's algorithm for determining the error values. The description of each of these blocks is out of scope of this paper, and the reader is referred to [5]-[10] for details and other references. It was found that the energy consumption of these blocks is given as follows:

$$\begin{aligned} \mathcal{E}_{sc} &= N(\mathcal{E}_{mult} + \mathcal{E}_{add}) \sum_{i=0}^{2T-1} \alpha_{sc,i} \\ \mathcal{E}_{bm} &= (10\mathcal{E}_{mult} + 6\mathcal{E}_{add}) \left(\sum_{i=0}^{T-1} \alpha_{bm,i} \right)^2 + 2\mathcal{E}_{inv} \sum_{i=0}^{T-1} \alpha_{bm,i} \\ \mathcal{E}_{cf} &= (2N(\mathcal{E}_{mult} + \mathcal{E}_{add}) + \mathcal{E}_{inv}) \sum_{i=0}^{T-1} \alpha_{cf,i}. \end{aligned} \quad (3.7)$$

Adding \mathcal{E}_{sc} , \mathcal{E}_{bm} and \mathcal{E}_{cf} in (3.7), and employing (2.6), we get the energy consumption of the decoder per codeword,

$$\mathcal{E}_{dec} = (4tN + 10t^2)\mathcal{E}_{mult} + (4tN + 6t^2)\mathcal{E}_{add} + 3t\mathcal{E}_{inv}, \quad (3.8)$$

where, as before, we have removed the time variable n .

3.4. RS Codec

The energy consumption of the codec can be found by adding the energy consumption of the encoder and the decoder in (3.6) and (3.8), respectively. The energy consumption of the codec per codeword is given by,

$$\mathcal{E}_{codec} = (6tN + 6t^2)\mathcal{E}_{mult} + (6tN + 6t^2)\mathcal{E}_{add} + 3t\mathcal{E}_{inv}. \quad (3.9)$$

Since, $k = N - 2t$ information symbols (each with m bits) are transmitted via each codeword. The energy consumption of the codec for processing one information bit is given by,

$$\mathcal{E}_{SPA}(c) = \frac{\mathcal{E}_{codec}}{m(N-2t)}. \quad (3.10)$$

Thus, we have obtained an expression for $\mathcal{E}_{SPA}(c)$. In the next section, we present reconfiguration strategy for choosing $c_{opt}(s_i)$.

4. ENERGY-EFFICIENT RECONFIGURATION STRATEGY

In this section, we present an energy-efficient reconfiguration strategy and compute energy savings.

4.1. Reconfiguration Strategy

First, we detect the input state $s(n)$ defined by (2.1) and then we compute $c_{opt}(s_i)$ according to (2.8). We can compute E_b/N_o as the SNR across the slicer (see [3]-[4]), which determines the input state, $s(n) \in \{s_i\}$. For each state s_i , the optimum configuration $c_{opt}(s_i)$ is computed by substituting (2.10) and (3.10) in (2.8) and solving the resulting optimization problem. These pre-computed configurations are stored in the look-up table. It can be shown that if N_s is the number of states, then a ROM of size $N_s \times 6T$ is required. From the SMA energy consumption point of view, it is better to keep N_s as small as possible. Also, the size of the ROM can be reduced to $N_s \times \lceil \log_2 T \rceil$, if we instead store $t(n)$ and compute the actual configuration vector $c(n)$ in real-time.

For each change in configuration, the transmitter needs to be informed. This is because the generator polynomial coefficients g_i in Fig. 2 needs to be changed for each change in $t(n)$. The new coefficients can be obtained either from a ROM storing the coefficients for all possible values of $t(n)$ or by updating the coefficients via the strategy in [12].

4.2. Energy Savings

In this subsection, we employ the DAT-based reconfigurable RS codec as the outer transceiver digital communication system of Fig. 1. The desired BER is $J_o = 10^{-10}$. Further, we assume that the inner transceiver is fixed, and E_b/N_o varies from 7dB to 10dB, which is modeled by $N_s = 16$ states given by (2.1). Further, the states have Gaussian probability distribution (with nominal SNR equal to 8.5dB and standard deviation of 1dB). Further, the reconfigurable RS codec has symbol size $m = 8$, code length $N = 2^m - 1 = 255$, and starting root of the generator polynomial $b = 0$. The number of correctable errors $t(n)$ can vary from 0 to $T = 32$.

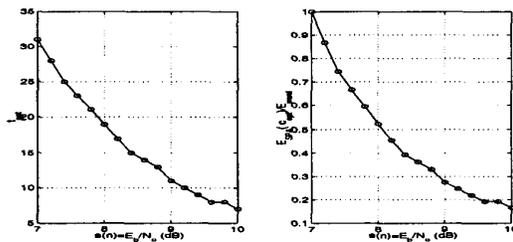


Figure 4: (a) t_{opt} and (b) $\mathcal{E}_{SPA}(c_{opt})/\mathcal{E}_{worst}$ versus $s(n)$.

Fig. 4(a) shows the variation of $t_{opt}(n)$ as the state $s(n)$ varies from s_1 to s_{16} . The state s_1 corresponds to the worst case for this example. The energy consumption of the SPA block for each of

the states is computed from (3.10), and is plotted in Fig. 4(b). The worst case energy dissipation is given by

$$\mathcal{E}_{worst} = \max_{s \in S} \mathcal{E}_{SPA}(c_{opt}(s)),$$

which corresponds to state s_1 . The energy consumption of the SMA block is dependent upon the transition probabilities of the input state, and the rate at which SPA block is reconfigured. If we assume that the reconfiguration rate is small as compared to the symbol rate, then the DAT energy dissipation is dominated by the SPA energy dissipation. The average energy savings are then computed as $\mathcal{E}_{sav} = (1 - \mathcal{E}_{DAT,ave}/\mathcal{E}_{worst})$. It was found that the energy savings range from 0% to 84% as the state changes from s_1 to s_{16} . On an average, we achieve 55% energy savings for this example. Future work includes extension to convolution codes, application to wireless channels, and ultimately joint optimization of the inner and outer receiver.

5. REFERENCES

- [1] K. K. Parhi, "Algorithm transformation techniques for concurrent processors," *Proceedings of the IEEE*, vol. 77, no. 12, pp. 1879-1895, Dec. 1989.
- [2] A. Chandrakasan *et al.*, "Minimizing power using transformations," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 1, pp. 12-31, Jan. 1995.
- [3] M. Goel and N. R. Shanbhag, "Low-power equalizers for 51.84 Mb/s very high-speed digital subscriber loop (VDSL) modems," in *Proceedings of IEEE VLSI Signal Processing Workshop*, (Boston, MA), Oct. 1998.
- [4] M. Goel and N. R. Shanbhag, "Dynamic algorithm transformations (DAT) for low-power adaptive signal processing," *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 161-166, Aug. 1997.
- [5] A. Raghupathy and K. J. R. Liu, "Low power/high speed design of a Reed Solomon decoder," in *ISCAS*, (Hong Kong), pp. 2060-2063, June 1997.
- [6] S. K. Jain, L. Song, and K. K. Parhi, "Efficient semisystolic architectures for finite-field arithmetic," *IEEE Trans. VLSI*, vol. 6, no. 1, pp. 101-113, 1998.
- [7] Y. R. Shayan and T. Le-Ngoc, "A cellular structure for a versatile Reed-Solomon decoder," *IEEE Trans. on Computers*, vol. 46, no. 1, pp. 80-85, Jan. 1997.
- [8] W. Wilhelm, A. Kaufman, and T. G. Noll, "A new scalable VLSI architecture for Reed-Solomon decoders," in *Proceedings, Custom Integrated Circuits Conference*, (Santa Clara, California), pp. 2.3.1-2.3.4, May 1998.
- [9] T. Le-Ngoc, M. T. Vo, B. Mallett, and V. K. Bhargava, "A gate-array based programmable Reed-Solomon codec: Structure - implementation - applications," in *Proceedings of IEEE Custom Integrated Circuits Conference*, pp. 121-124, 1990.
- [10] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1983.
- [11] M. G. Xakellis and F. N. Najm, "Statistical estimation of the switching activity in digital circuits," in *Design Automation Conference*, pp. 728-733, June 1994.
- [12] M. A. Hasan and V. K. Bhargava, "Architecture for a low-complexity rate-adaptive Reed-Solomon encoder," *IEEE Trans. on Computers*, vol. 44, no. 7, pp. 938-942, July 1995.