

Information-Theoretic Bounds On Average Signal Transition Activity

Sumant Ramprasad, Naresh R. Shanbhag, *Senior Member, IEEE*, and Ibrahim N. Hajj, *Fellow, IEEE*

Abstract—Transitions on high capacitance busses in VLSI systems result in considerable system power dissipation. Therefore, various coding schemes have been proposed in the literature to encode the input signal in order to reduce the number of transitions. In this paper, we derive lower and upper bounds on the average signal transition activity via an information-theoretic approach in which symbols generated by a process (possibly correlated) with *entropy rate* \mathcal{H} are coded with an average of R bits per symbol. The bounds are asymptotically achievable if the process is stationary and ergodic. We also present a coding algorithm based on the Lempel-Ziv data compression algorithm to achieve the bounds. Bounds are also obtained on the expected number of 1's (or 0's). These results are applied to, 1.) determine the activity reducing efficiency of different coding algorithms such as Entropy coding, Transition signaling, and Bus-Invert coding, and 2.) determine the lower-bound on the power-delay product given \mathcal{H} and R . Two examples are provided where transition activity within 4% and 9% of the lower bound is achieved when blocks of 8 symbols and 13 symbols, respectively, are coded at a time.

Keywords—Low power, switching activity, achievable bounds, CMOS circuits, information theory, busses

I. INTRODUCTION

Power dissipation has become a critical VLSI design concern in recent years [3] and a substantial amount of research is being conducted at the algorithmic [3], architectural (such as pipelining [13] and parallel processing) logic [9, 18] and circuit [4, 8] levels in order to develop power reduction techniques. Most of these efforts focus upon reducing the on-chip dynamic power dissipation of CMOS circuits, which at a node is given by,

$$P_D = \frac{1}{2}TC_LV_{dd}^2f, \quad (1.1)$$

where T is the transition activity at the node, C_L is the capacitance (the product TC_L represents the total switching capacitance), V_{dd} is the supply voltage, and f is the frequency of operation. At the system level, off-chip busses have capacitances, C_L , that are orders of magnitude greater than those found on signal lines internal to a chip. Therefore, transitions on these busses result in considerable system power dissipation. To address this problem, various signal encoding techniques have been proposed in the literature [2, 7, 14, 19, 20, 23] to encode the data before transmitting it on a bus so as to reduce the expected and the

peak number of transitions. Hence, the signal encoding approaches in [2, 7, 14, 19, 20, 23] achieve power reduction by reducing T in (1.1) while keeping C_L more or less unaltered. In this paper, we will explore the limits to which signal coding can be employed for the purpose of power reduction.

The work in [23] exploits the fact that the data transfers on microprocessor address busses are often sequential (i.e., current data value equals the previous data value plus a constant increment) due to instruction fetches. Hence, transition activity can be reduced by employing Gray codes, where only one bit changes between two successive codewords. Another encoding algorithm for address busses has been presented in [2]. In this algorithm, if the next address is greater than the current address by an increment of one then the bus lines are not altered and an extra "increment" bit is set to one. In [7], Fletcher presents an algorithm, termed Bus-Invert coding in [20], to reduce the number of transitions on a bus. This algorithm determines the number of bus lines that normally change state when the next output word is clocked onto the bus. When the number of transitions exceeds half the bus width, the output word is inverted before being clocked onto the bus. An extra output line is employed to signal the inversion. The Bus-Invert coding algorithm performs well when the data is uncorrelated. In [14], a two-step framework to reduce transition activity is presented in which data is passed through a decorrelating function f_1 , followed by a variant of entropy coding function, f_2 , which reduces the transition activity. The transition activity reducing algorithms have an analogue in the area of optical communications where the power dissipation depends on the number of ON (1) bits. In [6], Faulkner presents an encoding algorithm to reduce power dissipation in optical circuits by assigning codewords with fewer 1's to signal samples having a higher probability of occurrence.

In this paper, we derive lower and upper bounds on the average signal transition activity for any coding algorithm. These bounds are derived via an information-theoretic approach in which each symbol of a process (possibly correlated) with *entropy rate* \mathcal{H} is coded with R bits on average. The bounds are asymptotically achievable if the process is stationary (i.e., signal statistics such as mean do not change with time) and ergodic (i.e., the time average and ensemble average are equal). The transition reduction efficiency of existing coding algorithms are compared with the bounds derived in this paper. This work is a continuation of our effort in developing an information-theoretic view of VLSI computation [15], whereby equivalence between computation and communication is being established. This equiv-

Manuscript received July 8, 1997, revised December 9, 1997, April 14, 1998, September 10, 1998, and December 21, 1998

This work was supported by National Science Foundation CAREER award MIP 96-23737, DARPA contract DABT63-97-C-0025, and National Science Foundation award MIP 97-10235.

The authors are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. E-mail: {ramprasa, shanbhag, hajj}@uivlsi.csl.uiuc.edu.

IEEE Log Number 9800219

alence has provided lower bounds on power dissipation for digital VLSI systems [16, 17] and has for the first time provided a common thread linking various levels of the VLSI design hierarchy.

The concept of *entropy* from information theory was employed in the area of high-level power estimation in [10, 12]. In [12], entropy was employed as a measure of the average activity to be expected in the final implementation of a circuit, given only its Boolean functional description. In [10], information theory is employed to estimate power dissipation at logic and register-transfer levels. First, the output entropy of a circuit is estimated from the input entropy. Next, the entropy per circuit line is calculated from the input and output entropy values and used as an estimate for the average switching activity. In [10], the expected transition activity of a 1-bit signal is shown to be upper-bounded by one-half of its entropy under the temporal independence assumption and assuming level signaling. In contrast, the work presented here is applicable to multi-bit signals, independent of the coding algorithm, and completely unravels the connection between the bounds on transition activity and *entropy rate*. Also, the focus of this paper is not to estimate average switching activity, but to provide information-theoretic bounds on average switching activity.

In section II, we present the preliminaries necessary for the development in the rest of the paper. In section III, the main result is presented in the form of *Theorem 1*. In section III, we also present a coding algorithm that asymptotically achieves the lower bound for stationary and ergodic processes. In section IV, we employ Theorem 1 to, 1.) derive lower and upper bounds for different coding algorithms, and 2.) determine the lower-bound on the power-delay product. We also present two examples where transition activity within 4% and 9% of the lower bound is achieved.

II. PRELIMINARIES

In this section, we define terms employed in the rest of the paper. Let X be a discrete random variable with alphabet \mathcal{X} and probability mass function $p(x) = \Pr(X = x)$, $x \in \mathcal{X}$. A measure of the information content of X is given by its *entropy* $H(X)$, which is defined as follows [5],

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \text{ bits.} \quad (2.1)$$

This definition of the measure of information implies that the greater the *uncertainty* in the source output, the higher is its information content. In a similar fashion, a source with zero uncertainty would have zero information content and therefore its entropy would identically be equal to zero (from (2.1)).

The *joint entropy* $H(X_1, X_2, \dots, X_n)$ of a collection of discrete random variables (X_1, X_2, \dots, X_n) with a joint distribution $p(x_1, x_2, \dots, x_n)$ is defined as,

$$H(X_1, X_2, \dots, X_n) = - \sum p(x_1, x_2, \dots, x_n) \log_2 p(x_1, x_2, \dots, x_n) \quad (2.2)$$

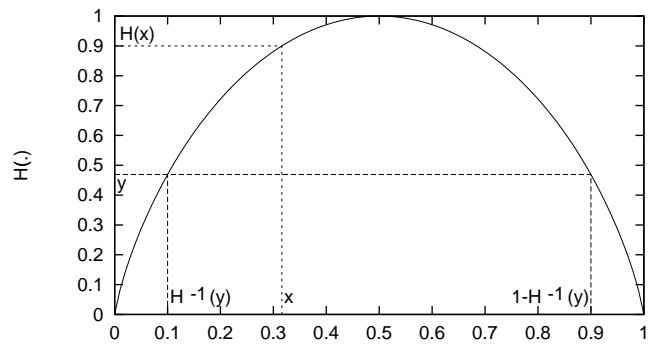


Fig. 1. A plot of the function $H(\cdot)$

The *entropy rate* of a stochastic process $\{X_i\}$ is defined as,

$$\mathcal{H} = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n) \text{ bits,} \quad (2.3)$$

when the limit exists. For an independent, identically distributed (i.i.d.) process, the entropy rate is equal to the entropy.

The function $H(x)$ is defined on the real interval $[0, 1]$ as follows,

$$H(x) = -x \log_2 x - (1-x) \log_2 (1-x) \text{ bits.} \quad (2.4)$$

The function $H(x)$, shown in Figure 1, maps the probability of a binary-valued discrete random variable to its entropy and has the following properties,

1. $H(0)$ and $H(1)$ are both defined to be 0
2. $H(x) = H(1-x)$
3. $H(x)$ is a concave function, i.e., a line drawn between any two points on the curve will lie below the curve. This is also referred to as Jensen's inequality: $\frac{1}{n} \sum_{i=1}^n H(x_i) \leq H(\frac{1}{n} \sum_{i=1}^n x_i)$.
4. The derivative of $H(x)$ with respect to x , $H'(x) = \log_2 \frac{1-x}{x}$
5. $H(x)$ is monotonically increasing in the interval $(0, \frac{1}{2}]$ because $H'(x) \geq 0$ in the interval with the equality occurring only at $x = \frac{1}{2}$
6. $H(x)$ is monotonically decreasing in the interval $[\frac{1}{2}, 1)$ because $H'(x) \leq 0$ in the interval with the equality occurring only at $x = \frac{1}{2}$

The inverse, $H^{-1}(y)$, of $H(\cdot)$, is defined on the real interval $[0, 1]$ as follows,

$$H^{-1}(y) = x, \text{ if } y = H(x) \text{ and } x \in [0, \frac{1}{2}]. \quad (2.5)$$

The function $H^{-1}(\cdot)$ maps the entropy of a binary-valued discrete random variable to a probability value that lies between 0 and $\frac{1}{2}$. Following the convention in literature, we denote three different functions as $H(\cdot)$. In (2.1) the argument of $H(\cdot)$ is a single random variable, in (2.2) the argument of $H(\cdot)$ is a sequence of random variables, and in (2.4) the argument of $H(\cdot)$ is a real number between 0 and 1. It will be clear from the context which function we are referring to.

The *transition activity* of a bit-level signal, b_n , is defined as [11],

$$t = \Pr(b_n = 0 \text{ and } b_{n-1} = 1) + \Pr(b_n = 1 \text{ and } b_{n-1} = 0).$$

We define *level signaling* to be the case where the bit ‘0’ is used to transmit the symbol ‘0’ and the bit ‘1’ is used to transmit the symbol ‘1’. In *transition signaling*, the symbol ‘0’ is transmitted by sending the previous transmitted bit and the symbol ‘1’ is transmitted by sending the complement of the previous transmitted bit. Hence, a ‘1’ is signaled with a transition and a ‘0’ with no transition.

III. BOUNDS ON SIGNAL TRANSITION ACTIVITY

In this section, we present the main results of this paper; namely, lower and upper bounds on the expected number of transitions. The bounds are asymptotically achievable if the process is stationary and ergodic. We also present a coding algorithm that asymptotically achieves the bounds. The proofs are presented in the Appendices.

In order to derive bounds on transition activity, we will employ Lemmas 1 and 2 presented below. Lemma 1 bounds x given $y \leq H(x)$. Lemma 2 employs Lemma 1 to bound the expected number of 1’s in a sequence of bits with a certain entropy rate. Theorem 1 employs Lemma 2 to bound the number of transitions per symbol of a process with a certain entropy rate given that each symbol is coded employing an expected number of R bits.

Lemma 1: *For all (x, y) such that $x \in [0, 1]$ and $y \in [0, 1]$, if $y \leq H(x)$ then,*

$$H^{-1}(y) \leq x \leq 1 - H^{-1}(y). \quad (3.1)$$

The relation between x and y in Lemma 1 is shown in Figure 1.

Lemma 2: *If,*

1. $\{B_i\}$ is a 0-1 valued stationary and ergodic process with entropy rate greater than or equal to \mathcal{H} ,
2. $p_i = \Pr(B_i = 1)$, and
3. $p_b = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n p_i$ exists then,

$$H^{-1}(\mathcal{H}) \leq p_b \leq 1 - H^{-1}(\mathcal{H}), \quad (3.2)$$

and the bounds in (3.2) are asymptotically achievable.

Theorem 1: *Let,*

1. \mathcal{H} be the entropy rate of a stationary and ergodic process $\{X_i\}$,
2. the symbols be coded in a uniquely decodable manner into bits represented by the binary random variables B_1, B_2, \dots employing an expected number of $R (> \mathcal{H})$ bits/symbol,
3. the bits be transmitted in some arbitrary manner over a finite set of wires such that a receiver can uniquely decode the bits, and
4. T be the expected number of transitions in the bits on the wires per symbol (i.e. $\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n B_i \oplus B_{\text{prev}(i)}$ exists, where the function $\text{prev}(i)$ returns the index of the bit that immediately precedes B_i on the same wire and \oplus is the exclusive-or operator)

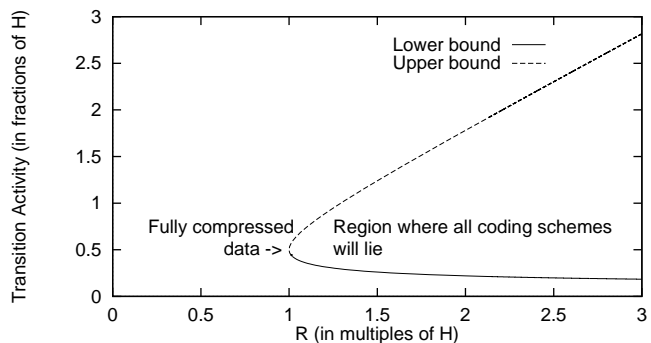


Fig. 2. Lower and Upper bounds on Transition Activity versus R

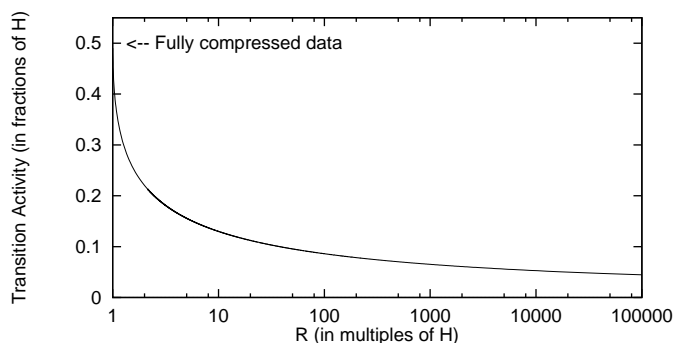


Fig. 3. Lower bound on Transition Activity versus R

then,

$$H^{-1}\left(\frac{\mathcal{H}}{R}\right)R \leq T \leq \left(1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right)\right)R, \quad (3.3)$$

and the bounds in (3.3) are asymptotically achievable.

In (3.3), $H^{-1}\left(\frac{\mathcal{H}}{R}\right)$ is the lower bound on the bit-level transition activity (this will be proved in Appendix C) which is scaled by R to give the lower bound on the expected number of transitions per symbol since R bits (on average) are employed to encode a symbol. The lower and upper bounds on transition activity computed by Theorem 1 for different values of R are shown in Figure 2. Any coding algorithm (such as those in [2, 7, 14, 19, 20, 23]) will need to reside in the region shown in Figure 2.

The lower bound on transition activity computed by Theorem 1 for different values of R is shown on a larger scale in Figure 3, where we see that the transition activity can be made arbitrarily close to 0 by increasing R . In practice, however, R will typically be less than approximately $10\mathcal{H}$ because most of the reduction in the lower bound is achieved by the time R is equal to $10\mathcal{H}$.

Theorem 1 also indicates that a coding algorithm that achieves the lower bound on transition activity can be modified to achieve the upper bound by mapping a transition to a no transition and a no transition to a transition. This is possible, since the sum of probabilities of a transition and of no transition is unity, the bit-level transition activity of such a coding algorithm is $1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right)$, thereby achieving the upper bound in (3.3).

Remark 1: The bounds on the expected number of 1’s

per symbol, P , are the same as those for T , i.e.,

$$H^{-1}\left(\frac{\mathcal{H}}{R}\right)R \leq P \leq (1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right))R. \quad (3.4)$$

The proof of (3.4) is similar to that of Theorem 1.

A. An Asymptotically Optimal Coding Algorithm

We now present a modified Lempel-Ziv (MLZ) algorithm, that is asymptotically optimal in that it achieves the lower bound on transition activity for a stationary and ergodic process. The purpose of presenting this algorithm is to show how a widely used data compression algorithm can be modified to approach the lower bound in (3.3). While the MLZ algorithm will minimize T in (3.3), it may not minimize power dissipation because of the increase in C_L due to the encoding and decoding process. The MLZ algorithm has the same resource requirements as the original Lempel-Ziv algorithm [25], and is universal, i.e., it does not require the source probability distribution be computed. Without loss of generality, we will assume that the source alphabet consists of only two symbols ‘0’ and ‘1’. The MLZ algorithm accepts the average number of bits per symbol, R , as a parameter.

1. Pass 1: The source sequence is sequentially parsed into phrases that have not appeared so far. For example, if the string is 1011010100010..., we parse it as 1,0,11,01,010,00,10,... After every comma, we look along the input sequence until we come to the shortest phrase that has not been marked off before. Since this is the shortest such phrase, the phrase consisting of all but the last bit of this phrase must have occurred earlier. In pass 2, we code this phrase by giving the location of the prefix and the value of the last bit.
2. Pass 2: Let $c(n)$ be the number of phrases in the parsing of the input n -sequence. We employ nR bits in total to code the n input bits. Hence, the number of bits available to describe the location of the prefix is $\frac{nR}{c(n)} - 1$ and 1 bit to describe the last bit. We choose the addresses in order of increasing number of 1’s, i.e., the all 0’s pattern, followed by words with exactly one ‘1’, followed by words with exactly two 1’s, and so on. For example, the code for the above sequence (assuming the number of bits, $\frac{nR}{c(n)} - 1$, employed to code the location of the prefix is three) is (000,1)(000,0)(001,1)(010,1)(011,0)(010,0)(001,0),... where the first number of each pair gives the index of the prefix and the second number gives the last bit of the phrase. Decoding the coded sequence is straightforward and we can recover the source sequence without error.
3. A ‘1’ is transmitted with a transition and a ‘0’ with no transition.

The algorithm can be modified so that it requires only one pass over the string without affecting the asymptotic efficiency of the algorithm [1, 24]. The proof of asymptotic optimality of the MLZ algorithm is presented in Appendix D. There are two differences between the MLZ algorithm and the original Lempel-Ziv algorithm,

1. the number of bits allocated to the pointers is $\frac{nR}{c(n)} - 1$ in the MLZ algorithm versus $\log_2 c(n)$ in the original Lempel-Ziv algorithm, and
2. the number representation employed in the pointers is increasing number of 1’s in the MLZ algorithm versus unsigned number representation in the original algorithm.

IV. APPLICATIONS OF BOUNDS ON TRANSITION ACTIVITY

Theorem 1 is useful in designing coding schemes because knowledge of the lower bound will tell the designer how close to the optimum a given coding scheme is, and whether it is worthwhile to search for a better coding scheme. In addition, the proofs of asymptotic achievability provide ideas on how good practical coding schemes may be designed. We have used these ideas to come up with practical coding schemes. These coding schemes are not described in this paper due to space constraints, but are described in [14].

In this section, we employ Theorem 1 and the value of R specified by a coding algorithm to derive bounds for that algorithm. We also apply Theorem 1 to determine the lower bound on the power-delay product. In addition, we present two examples to illustrate Theorem 1.

A. Fully compressed data

In fully compressed data, the expected number of bits per symbol is equal to the entropy rate of the process.

Corollary 1: For fully compressed data, $T = \frac{\mathcal{H}}{2}$.

Proof:

$$\begin{aligned} H^{-1}\left(\frac{\mathcal{H}}{R}\right)R &\leq T \leq (1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right))R \quad [(3.3)] \\ \Rightarrow H^{-1}(1)\mathcal{H} &\leq T \leq (1 - H^{-1}(1))\mathcal{H} \\ &[\text{Since } R = \mathcal{H} \text{ for fully compressed data}] \\ \Rightarrow T &= \frac{\mathcal{H}}{2} \quad [\text{Since } H^{-1}(1) = \frac{1}{2}]. \quad \blacksquare \end{aligned}$$

Therefore, as shown in Figure 2, the lower and upper bounds on T are identical for fully compressed data.

B. Transition Signaling

Consider a source with alphabet comprising of two symbols, ‘0’ and ‘1’. We will redefine transition signaling slightly in this sub-section: we will signal the less probable symbol (‘0’ or ‘1’) with a transition and the other symbol with no transition.

Corollary 2: Transition signaling achieves the lower bound on the transition activity for an i.i.d. source with alphabet $\mathcal{X} = \{0, 1\}$ when $R = 1$ bit/symbol.

Proof:

$$\begin{aligned} H^{-1}\left(\frac{\mathcal{H}}{R}\right)R &\leq T \leq (1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right))R \quad [(3.3)] \\ \Rightarrow H^{-1}(\mathcal{H}) &\leq T \leq 1 - H^{-1}(\mathcal{H}) \quad [\text{Since } R = 1] \end{aligned}$$

Since the source is i.i.d. with alphabet $\{0,1\}$, $H^{-1}(\mathcal{H})$ is also the probability of a ‘0’ or the probability of a ‘1’, whichever is less. Hence we can achieve the lower bound on

the transition activity by signaling the less probable symbol ('0' or '1') with a transition and the other symbol with a no transition. ¶

The upper bound on transition activity of $1 - H^{-1}(\mathcal{H})$ is greater than or equal to the upper bound of $\frac{\mathcal{H}}{2}$ in [10]. This is because, in the proof of the upper bound in [10], the implicit assumption is made that the symbol '0' is signaled with the bit '0' and the symbol '1' is signaled with the bit '1', i.e., level signaling is employed. It is possible to achieve the higher transition activity of $1 - H^{-1}(\mathcal{H})$ for the same entropy rate if the more probable symbol is signaling with a transition.

C. Bounds For 1-bit Redundant Codes

In algorithms such as Bus-Invert coding [7, 20], the transition activity on the bus is reduced by employing an additional bit. We now calculate the lower bound for any coding algorithm that uses 1 bit of redundancy. Thus $R = \mathcal{H} + 1$ and hence, from Theorem 1, the expected transition activity is bounded by,

$$(\mathcal{H} + 1)H^{-1}\left(\frac{\mathcal{H}}{\mathcal{H} + 1}\right) \leq T \leq (\mathcal{H} + 1)\left(1 - H^{-1}\left(\frac{\mathcal{H}}{\mathcal{H} + 1}\right)\right). \quad (4.1)$$

The extra bit may be either an extra line on the bus, or an extra clock to transfer the data [19]. If $\mathcal{H} = 8$ then the lower bound from (4.1) is 2.7569 transitions/symbol. For uniformly distributed, temporally independent data, Bus-Invert coding achieves a transition activity of 3.273 transitions/symbol which is 18.72% above the lower bound. The lower bound in (4.1) can be approached by coding larger and larger blocks of bits. Now, assume the entropy rate, \mathcal{H} , of the process is increased and $R = \mathcal{H} + 1$. The ratio $\frac{\mathcal{H}}{\mathcal{H} + 1}$ in (4.1) approaches 1 and T approaches $\frac{\mathcal{H}}{2}$. Thus, as the entropy rate increases, the benefit of Bus-Invert coding or any 1-bit redundant code is reduced. The above analysis can also be extended for a k -bit redundant code.

D. Lower Bound On Power-Delay Product

If the capacitance C_L , the supply voltage V_{dd} , and the frequency of operation, f , are given, then the minimum average power dissipation is proportional to the lower bound on the transition activity. If the number of wires employed to transmit the data is fixed, the transmission delay is proportional to R . For instance, if R is doubled and the bus width is unchanged, then twice as many clock cycles are needed to transmit the same number of symbols. Hence the lower bound on the power-delay product, $PowerDelay_{min}$, given \mathcal{H} and R , is given by,

$$PowerDelay_{min} = KH^{-1}\left(\frac{\mathcal{H}}{R}\right)R^2, \quad (4.2)$$

where K is a constant of proportionality. The graph of $PowerDelay_{min}$ versus R for a given value of \mathcal{H} is shown in Figure 4. For given \mathcal{H} , we can find the R that minimizes $PowerDelay_{min}$ by equating the derivative of (4.2) with respect to R , to 0. The value of R that minimizes $PowerDelay_{min}$ is found to be,

$$R_{min, power-delay} = 1.25392 \mathcal{H}. \quad (4.3)$$

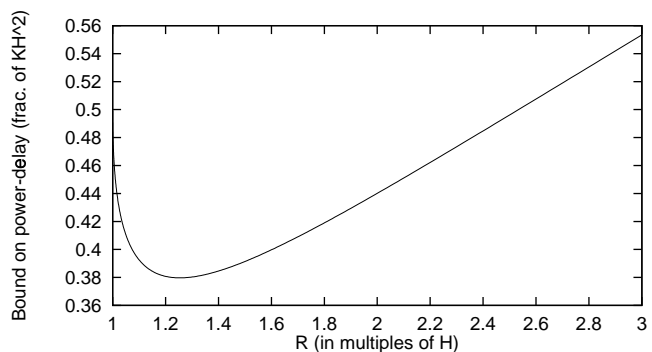


Fig. 4. Lower bound on power-delay versus R for given \mathcal{H}

Thus, (4.3) indicates that a process with entropy rate, \mathcal{H} , requires approximately an average of $1.25\mathcal{H}$ bits per symbol to encode for minimum power-delay product. If $R > 1.25\mathcal{H}$, then the delay component will increase resulting in a non-optimal power-delay product. Similarly, if $R < 1.25\mathcal{H}$ then the power component increases because less redundancy is being added.

In order to illustrate Theorem 1, we now calculate bounds for two different types of sources and the transition activity for coding algorithms that approach the bounds.

E. Bounds On Transition Activity For An i.i.d. Source

Consider an i.i.d. source with a 5 symbol alphabet $\mathcal{X} = \{A, B, C, D, E\}$ with probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$, and $\frac{1}{16}$ respectively. Since the source is i.i.d., the entropy rate is equal to the entropy and is given by, $\mathcal{H} = \frac{15}{8}$ bits. Assume an average of $R = 3$ bits are employed to code a symbol. Thus $\frac{\mathcal{H}}{R} = \frac{5}{8}$, $H^{-1}\left(\frac{\mathcal{H}}{R}\right) = 0.156142$, and from Theorem 1, the bounds on transition activity are,

$$0.468426 \leq T \leq 2.531574. \quad (4.4)$$

We now calculate the actual transition activity that is achieved by various coding algorithms and compare them with the bounds in (4.4). To simplify the calculation of transition activity, we assume transition signaling is employed to transmit the bits, i.e., a '1' is transmitted with a transition and a '0' is transmitted with no transition. Thus, the number of transitions is equal to the number of 1's. We can make the assumption of transition signaling in the examples because the purpose of the examples is to show the existence of coding algorithms that approach the lower bound.

E.1 Entropy Coding Followed By Spatial Redundancy Coding

In this algorithm, we initially employ an entropy coder to code the symbols employing the minimum expected number of bits per symbol. A code that achieves the entropy is A = 0, B = 10, C = 110, D = 1110, and E = 1111. The output of the entropy coder consists of temporally independent, uniformly distributed bits. These bits are placed in a buffer and when there are 15 bits in a buffer, then the bits are encoded employing 24 bits. Since $\mathcal{H} = \frac{5}{8}$, the 15 bits

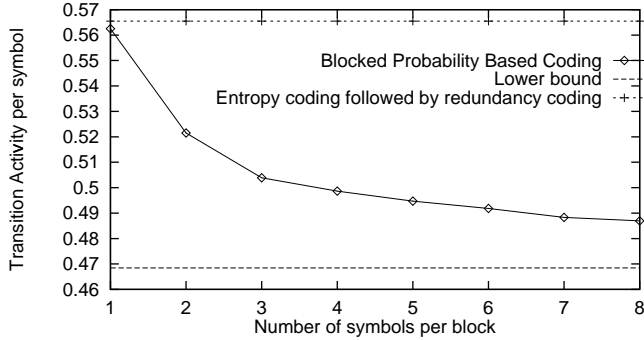


Fig. 5. Transition Activity versus Block Size for i.i.d. source

TABLE I
TRANSITION MATRIX

Previous state	S_1	S_2	S_3
S_1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$
S_2	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
S_3	0	$\frac{1}{2}$	$\frac{1}{2}$

correspond to an average of 8 symbols and hence employing 24 bits results in an expected number of 3 bits/symbol. Since we are encoding 15 bits employing 24 bits, we can employ the redundancy to reduce the number of transitions [19]. This is an extension of Bus-Invert coding and results in a 5-limited weight code [21, 22] in which we employ the all-zero (or no transitions) code, 24 codes with a single one, 276 codes ($= \binom{24}{2}$) with two ones, etc., up to 19817 of the codes with 5 ones. The expected number of transitions (or 1's) in each transmission of 24 bits is given by [14], $\frac{\sum_{i=0}^4 i \cdot \binom{24}{i} + 5 \cdot (2^{15} - \sum_{i=0}^4 \binom{24}{i})}{2^{15}} = 4.52383$ transitions. Since 24 bits correspond to an average of 8 symbols, the expected number of transitions per symbol is given by, $\frac{1}{8} * 4.52383 = 0.565478$ transitions/symbol, which is 20.7% above the lower bound in (4.4).

E.2 Probability Based Coding

An alternative algorithm to reduce the number of transitions is to code the most probable symbol A as 000 or no transitions, B as 001, C as 010, D as 100, and E as 011. The expected number of transitions per symbol is, $T = 0.5625$ transitions/symbol, which is 20.1% above the lower bound in (4.4). We can further reduce the number of transitions by applying probability based coding to a block of two symbols. and can be reduced further by coding with block sizes larger than two. The transition activity per symbol for different block sizes is shown in Figure 5. Thus, we can achieve a transition activity within 4% of the lower bound by employing a block size of 8.

F. Bounds On Transition Activity For A Markov Process

Consider the 3-state stationary Markov process U_1, U_2, \dots having the transition matrix P_{ij} in Table I [5]. Thus the probability that S_1 follows S_3 is equal to zero. An algorithm to encode the process will consist of 3 codes $C_1,$

TABLE II
ENTROPY CODE FOR MARKOV PROCESS

	S_1	S_2	S_3
C_1	0	10	11
C_2	10	0	11
C_3	—	0	1

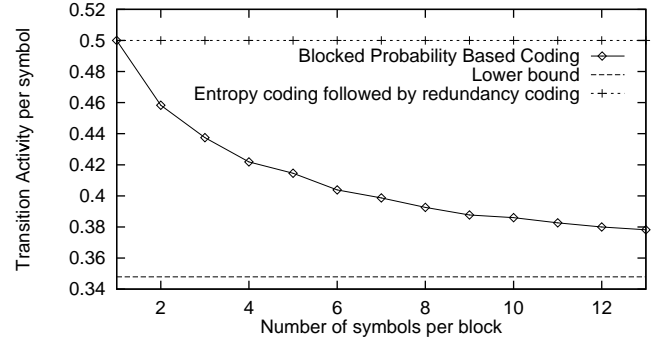


Fig. 6. Transition Activity versus Block Size for Markov Process

$C_2,$ and C_3 (one for each state $S_1, S_2,$ and S_3), where C_i is a code mapping from elements of the set $\{S_1, S_2, S_3\}$ into a codeword in C_i (see Table II for an example). The following algorithm will be employed to encode this Markov chain,

1. Note the present symbol S_i .
2. Select code C_i .
3. Note the next symbol S_j and send the codeword in C_i corresponding to S_j .
4. Repeat for the next symbol.

The stationary distribution of this Markov chain is $\mu = [\frac{2}{9}, \frac{4}{9}, \frac{1}{3}]^T$. The entropy rate of the stationary Markov process is given by [5],

$$H(\mathcal{X}) = - \sum_{ij} \mu_i P_{ij} \log_2 P_{ij} = \frac{4}{3} \text{ bits.} \quad (4.5)$$

A code that achieves the entropy rate is shown in Table II. If the symbols are coded with an average of $R = 2$ bits/symbol, then $\frac{\mathcal{H}}{R} = \frac{2}{3}$, $H^{-1}(\frac{\mathcal{H}}{R}) = 0.173952$, and from Theorem 1,

$$0.347904 \leq T \leq 1.652096 \text{ transitions/symbol.} \quad (4.6)$$

The transitions/symbol for entropy coding followed by redundancy coding and probability based coding is shown in Figure 6 for different block sizes. We can achieve a transition activity within 9% of the lower bound with a block size of 13 symbols.

To summarize, the above examples demonstrate that for the specific source and source distribution,

- We can achieve transition activities within 4% and 9% of the lower bound with block sizes of 8 symbols and 13 symbols, respectively (see Figure 5 and Figure 6).
- Transition activity can be reduced by coding larger and larger blocks of symbols (see Figure 5 and Figure 6).

- Entropy coding followed by redundancy coding does not always result in the minimum transition activity for a given block size.

V. CONCLUSIONS

In this paper, we have presented a new, non-empirical relation between the transition activity per symbol, T , and the entropy rate, \mathcal{H} , of a process by deriving lower and upper bounds on the expected transition activity per symbol and the expected number of 1's per symbol given the entropy rate \mathcal{H} of a process and the expected number of bits R employed to code a symbol. The bounds are asymptotically achievable if the process is stationary and ergodic. We also presented a coding algorithm based on the Lempel-Ziv compression algorithm to achieve the bounds. We employed the theoretical results to, 1.) derive lower and upper bounds on T for different coding algorithms, and 2.) determine the lower bound on the power-delay product given \mathcal{H} and R .

In future, we plan to apply Theorem 1 to bound the error in estimating switching activity for circuits employing entropy rate.

APPENDIX A

PROOF OF LEMMA 1

Consider two cases for y , ($0 < y \leq 1$) and $y = 0$:

Case 1: $0 < y \leq 1$: Thus $0 < H^{-1}(y) \leq \frac{1}{2}$. There are 3 cases for x :

- $x = 1$ or $x = 0$: This is not possible because $H(x) = 0$ and $y > 0$ which would violate $y \leq H(x)$
- $\frac{1}{2} \leq x < 1$:

$$H(1 - H^{-1}(y)) \leq H(x)$$

[Since $y = H(H^{-1}(y)) = H(1 - H^{-1}(y))$ and $y \leq H(x)$]

$$\Rightarrow x \leq 1 - H^{-1}(y)$$

[Property 6: $H(\cdot)$ is monotonically decreasing in $[\frac{1}{2}, 1]$]

$$\Rightarrow H^{-1}(y) \leq x \leq 1 - H^{-1}(y)$$

[Since $0 < H^{-1}(y) \leq \frac{1}{2}$ and $\frac{1}{2} \leq x < 1$]

- $0 < x < \frac{1}{2}$:

$$H^{-1}(y) \leq x$$

[Since $y \leq H(x)$ & $H(\cdot)$ is increasing in $(0, \frac{1}{2}]$]

$$\Rightarrow H^{-1}(y) \leq x < 1 - H^{-1}(y)$$

[Since $0 < H^{-1}(y) \leq \frac{1}{2}$ and $0 < x < \frac{1}{2}$]

Case 2: $y = 0$: Since $H^{-1}(y) = 0$ and $0 \leq x \leq 1$, (3.1) is satisfied

Hence the proof. \blacktriangleleft

APPENDIX B

PROOF OF LEMMA 2

From the definitions of entropy rate in (2.3) and \mathcal{H} in the statement of Lemma 2,

$$\mathcal{H} \leq \lim_{n \rightarrow \infty} \frac{1}{n} H(B_1, B_2, \dots, B_n),$$

$$\leq \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n H(B_i) \text{ [Independence bound on entropy]}$$

$$= \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n H(p_i) \text{ [} p_i = \Pr(B_i = 1)\text{]}$$

$$\leq H\left(\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n p_i\right)$$

[Property 3: Jensen's inequality and concavity of H],

$$\Rightarrow \mathcal{H} \leq H(p_b) \text{ [By definition, } p_b = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n p_i\text{]}$$

Thus, we can substitute \mathcal{H} for y and p_b for x in Lemma 1 to obtain (3.2). \blacktriangleleft

Proof Of Asymptotic Achievability Of Lemma 2:

We now present a coding algorithm, referred to as L2, that asymptotically achieves the lower bound in Lemma 2 for stationary and ergodic processes.

1. We encode each sequence of n symbols employing n code bits. We can do this because the source alphabet consists of only 2 symbols. The Asymptotic Equipartition Property (AEP) [5] states that given a stationary and ergodic process, for each $\epsilon_1 > 0$, there exists n_1 such that for all $n > n_1$ the following properties hold,

- (a) there is a set, called the typical set, $A_n^{\epsilon_1}$, which is a subset of the set of all possible sequences of n symbols generated by the process,
- (b) the number of elements in $A_n^{\epsilon_1}$, $|A_n^{\epsilon_1}|$, is bounded by,

$$(1 - \epsilon_1)2^{n(\mathcal{H} - \epsilon_1)} \leq |A_n^{\epsilon_1}| \leq 2^{n(\mathcal{H} + \epsilon_1)}, \quad (\text{B.1})$$

- (c) the probability of $A_n^{\epsilon_1}$ containing a sequence of n symbols generated by the process is *at least* $1 - \epsilon_1$, and
- (d) $\epsilon_1 \rightarrow 0$ as $n \rightarrow \infty$.

In short, AEP states that as the length of the sequence n increases, the probability that a generated sequence belongs to $A_n^{\epsilon_1}$ approaches unity, and the size of the typical set, $|A_n^{\epsilon_1}|$, approaches $2^{n\mathcal{H}}$.

2. We generate a set, $C_n^{\epsilon_2}$, of codewords. Each codeword in $C_n^{\epsilon_2}$ is formed by drawing n code bits in an independent, identically distributed manner with probability p of being a '1'. Again, from AEP, we know that the set $C_n^{\epsilon_2}$ will contain at least $(1 - \epsilon_2)2^{n(H(p) - \epsilon_2)}$ distinct codewords. We choose p such that the number of codewords in $C_n^{\epsilon_2}$ is at least the number of sequences in $A_n^{\epsilon_1}$, i.e.,

$$(1 - \epsilon_2)2^{n(H(p) - \epsilon_2)} = 2^{n(\mathcal{H} + \epsilon_1)}$$

$$\Rightarrow p = H^{-1}\left(\mathcal{H} + \epsilon_1 + \epsilon_2 - \frac{\log_2(1 - \epsilon_2)}{n}\right)$$

As $n \rightarrow \infty$, both ϵ_1 and $\epsilon_2 \rightarrow 0$, and $p \rightarrow H^{-1}(\mathcal{H})$.

3. We assign codewords to sequences in $A_n^{\epsilon_1}$ from the set $C_n^{\epsilon_2}$.
4. After each sequence in $A_n^{\epsilon_1}$ has been assigned a codeword from $C_n^{\epsilon_2}$, sequences not in $A_n^{\epsilon_1}$ are assigned

codewords in an arbitrary manner from the remaining codewords (which may or may not be in $C_n^{\epsilon_2}$).

As $n \rightarrow \infty$, the probability of a '1' at the output of the L2 encoder is at most $\frac{(1-\epsilon_1)pn + \epsilon_1 n}{n}$, where,

1. the probability of a sequence being in $A_n^{\epsilon_1}$ is at least $1 - \epsilon_1$ (AEP),
2. as $n \rightarrow \infty$, the number of 1's in a codeword encoding a sequence from $A_n^{\epsilon_1}$ is pn (strong law of large numbers),
3. the probability of a sequence not being in $A_n^{\epsilon_1}$ is at most ϵ_1 (AEP), and
4. the number of 1's in a codeword encoding a sequence not from $A_n^{\epsilon_1}$ is at most n (since length of codeword is n).

Hence as $n \rightarrow \infty$, the probability of a '1' at the output of the L2 encoder is p , or $H^{-1}(\mathcal{H})$, thereby achieving the lower bound in Lemma 2. The L2 coding algorithm can be modified to achieve the upper bound by exchanging 1's and 0's. \blacktriangleleft

APPENDIX C

PROOF OF THEOREM 1

Outline of proof: We will first prove that the entropy rate is not altered in the transition domain where a '1' represents a transition and a '0' represents no transition. We will then employ Lemma 2 to bound the number of 1's in the transition domain, which will in turn bound the number of transitions.

Proof: Consider any uniquely decodable coding scheme that codes the first N symbols, represented by the random variables (X_1, X_2, \dots, X_N) , generated by the process. The symbols are coded, independently of any other symbols, to the $n = NR$ bits represented by the binary random variables B_1, B_2, \dots, B_n and transmits these n bits in any order. Clearly, as N (or n) $\rightarrow \infty$, we are considering all possible uniquely decodable coding schemes that encode the process employing an expected number of R bits to code a symbol. Hence the bounds obtained as $n \rightarrow \infty$ will hold for all possible uniquely decodable coding schemes that encode the process employing an expected number of R bits to code a symbol.

For the bits to be uniquely decodable, the first N symbols must be some function f of the n bits the symbols are coded to, i.e.,

$$(X_1, X_2, \dots, X_N) = f(B_1, B_2, \dots, B_n). \quad (C.1)$$

The entropy rate of the process $\{B_i\}$ is given by,

$$\begin{aligned} & \lim_{n \rightarrow \infty} \frac{1}{n} H(B_1, B_2, \dots, B_n) \\ & \geq \lim_{n \rightarrow \infty} \frac{1}{n} H(f(B_1, B_2, \dots, B_n)) \\ & \text{[Joint entropy of a collection of random variables} \geq \text{Joint} \\ & \text{entropy of a function of the collection of random variables]} \\ & = \lim_{N \rightarrow \infty} \frac{1}{NR} H(X_1, X_2, \dots, X_N) \\ & = \frac{\mathcal{H}}{R} \end{aligned}$$

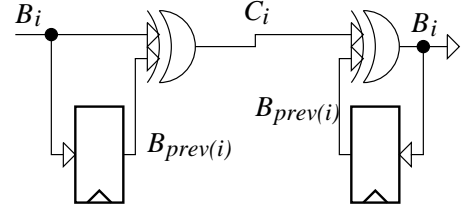


Fig. 7. Relation between B_i , $B_{prev(i)}$, and C_i

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{1}{n} H(B_1, B_2, \dots, B_n) \geq \frac{\mathcal{H}}{R}. \quad (C.2)$$

Define a function g on (B_1, B_2, \dots, B_n) as follows,

$$(C_1, C_2, \dots, C_n) = g(B_1, B_2, \dots, B_n), \quad (C.3)$$

where,

$$C_i = B_i \oplus B_{prev(i)}, \quad (C.4)$$

where the function $prev(i)$ returns the index of the bit that is transmitted on the same wire as B_i and immediately precedes B_i . If B_i is the first bit transmitted on the wire, then $B_{prev(i)}$ is '0'. Hence we can recursively compute (B_1, B_2, \dots, B_n) given (C_1, C_2, \dots, C_n) as follows,

$$B_i = C_i \oplus B_{prev(i)}. \quad (C.5)$$

Figure 7 shows the relation between B_i , $B_{prev(i)}$, and C_i . The two delays in Figure 7 are initialized to the same state so that C_i can be generated uniquely given B_i and vice versa making the function g bijective. Note that B_i and $B_{prev(i)}$ are transmitted on the same wire. If B_j is transmitted on a different wire from B_i , then we have another set of delays and exclusive-or gates for the generation of C_j from B_j and $B_{prev(j)}$. Since g is an invertible function,

$$\begin{aligned} & \lim_{n \rightarrow \infty} \frac{1}{n} H(C_1, C_2, \dots, C_n) = \\ & \lim_{n \rightarrow \infty} \frac{1}{n} H(B_1, B_2, \dots, B_n). \end{aligned} \quad (C.6)$$

From (C.2) and (C.6) we get,

$$\lim_{n \rightarrow \infty} \frac{1}{n} H(C_1, C_2, \dots, C_n) \geq \frac{\mathcal{H}}{R}. \quad (C.7)$$

Let $p_c = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n C_i$, where p_c is the probability of C_i being a '1'. The limit exists because of assumption (4) in the theorem. Since C_i is a binary random variable, $p_c n$ is the number of 1's in (C_1, C_2, \dots, C_n) for large n . Substituting p_c for p_b and $\frac{\mathcal{H}}{R}$ for \mathcal{H} in Lemma 2 we obtain,

$$H^{-1}\left(\frac{\mathcal{H}}{R}\right) \leq p_c \leq 1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right). \quad (C.8)$$

Since there are on the average R bits per symbol and C_i is '1' iff there was a transition at B_i ,

$$T = p_c R. \quad (C.9)$$

Multiplying (C.8) by R and employing (C.9) we have,

$$H^{-1}\left(\frac{\mathcal{H}}{R}\right)R \leq T \leq (1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right))R, \quad (C.10)$$

which is the desired result. \blacktriangleleft

Proof Of Asymptotic Achievability Of Theorem 1:

We now present a coding algorithm, T1, that asymptotically achieves the lower bound on transition activity in Theorem 1 for stationary and ergodic processes. The T1 coding algorithm is similar to the L2 algorithm with the differences being that the source is not binary and R is now not restricted to being 1. We provide an outline of the proof of asymptotic optimality of the T1 algorithm. The detailed proof is similar to the proof of asymptotic optimality of the L2 algorithm.

1. We first group blocks of k symbols from the source. For large k there are $2^{k\mathcal{H}}$ distinct blocks of symbols all of which are equally likely (AEP).
2. We code each block of k symbols employing kR bits. As in the L2 algorithm, each bit in a codeword encoding a block of symbols is chosen in an i.i.d. manner with probability $H^{-1}(\frac{\mathcal{H}}{R})$ of being a 1. Thus there are $2^{kRH(H^{-1}(\frac{\mathcal{H}}{R}))} = 2^{k\mathcal{H}}$ codewords (AEP). Hence we have a codeword for each block of symbols. Since each bit in each codeword has probability $H^{-1}(\frac{\mathcal{H}}{R})$ of being a 1, the bit-level probability at the output of the T1 encoder is also $H^{-1}(\frac{\mathcal{H}}{R})$.
3. In the last step we map a ‘1’ to a transition waveform and a ‘0’ to a transitionless waveform. Thus the bit-level transition activity is $H^{-1}(\frac{\mathcal{H}}{R})$ which is then scaled by R to achieve the lower bound on transitions per symbol. ¶

APPENDIX D**PROOF OF ASYMPTOTIC OPTIMALITY OF THE MLZ ALGORITHM**

Outline of proof: We first prove the asymptotic optimality of another algorithm MMLZ. The asymptotic optimality of the MLZ algorithm is obtained by showing that MLZ will be better than or equal to the MMLZ algorithm.

Proof: The modified MLZ algorithm (MMLZ) is identical to the MLZ algorithm except for pass 2 in sub-section III(A) where the bits in the prefix are chosen, not in order of increasing number of 1’s, but in an i.i.d. manner with probability p of being a ‘1’.

Since there are $\frac{nR}{c(n)} - 1$ bits in each prefix, from AEP we know that there are at least $(1 - \epsilon)2^{(\frac{nR}{c(n)} - 1)(H(p) - \epsilon)}$ distinct prefixes. We choose p such that the number of prefixes is at least equal to the number of phrases, $c(n)$, i.e.,

$$\begin{aligned} c(n) &= (1 - \epsilon)2^{(\frac{nR}{c(n)} - 1)(H(p) - \epsilon)} \\ \Rightarrow p &= H^{-1}\left(\frac{c(n) \log_2 c(n)}{nR(1 - \frac{c(n)}{nR})} - \frac{c(n) \log_2(1 - \epsilon)}{nR(1 - \frac{c(n)}{nR})} + \epsilon\right). \end{aligned}$$

As $n \rightarrow \infty$, $\epsilon \rightarrow 0$, and, from the proof of optimality of the original Lempel-Ziv algorithm [5], we know that,

$$\lim_{n \rightarrow \infty} \frac{c(n) \log_2 c(n)}{n} = \mathcal{H}, \quad (\text{D.11})$$

$$\lim_{n \rightarrow \infty} \frac{c(n)}{n} = 0. \quad (\text{D.12})$$

Hence,

$$\lim_{n \rightarrow \infty} p = H^{-1}\left(\frac{\mathcal{H}}{R}\right). \quad (\text{D.13})$$

As $n \rightarrow \infty$, the probability of a 1 at the output of the MMLZ encoder (which includes the prefixes and the additional bits), p_{mmlz} , is upper bounded by,

$$p_{mmlz} \leq \frac{c(n)((\frac{nR}{c(n)} - 1)p + 1)}{nR}. \quad (\text{D.14})$$

We can obtain (D.14) as follows,

- as $n \rightarrow \infty$, the number of 1’s in the prefix is $(\frac{nR}{c(n)} - 1)p$ (strong law of large numbers),
 - $(\frac{nR}{c(n)} - 1)p + 1$ is the maximum number of 1’s including the prefix and the additional bit,
 - $c(n)((\frac{nR}{c(n)} - 1)p + 1)$ is the maximum number of 1’s in the coded sequence which is then divided by the total number of bits, nR , to give the upper bound on p_{mmlz} .
- Simplifying (D.14), we obtain,

$$p_{mmlz} \leq \left(1 - \frac{c(n)}{nR}\right)p + \frac{c(n)}{nR}, \quad (\text{D.15})$$

which approaches p or $H^{-1}(\frac{\mathcal{H}}{R})$ as $n \rightarrow \infty$ (see (D.12)). Thus by selecting the bits in the prefix in an i.i.d. manner with probability $H^{-1}(\frac{\mathcal{H}}{R})$ we arrive at an asymptotically optimal algorithm. The MLZ algorithm in sub-section III(A), which selects the prefixes in order of increasing number of 1’s, will always be better than or equal to the MMLZ algorithm, in which all bits in the prefix have probability $H^{-1}(\frac{\mathcal{H}}{R})$ of being a 1. The reason for this is as follows,

1. Since the bits in the prefix are chosen in order of increasing number of 1’s in the MLZ algorithm, the prefixes in the MLZ algorithm will have 0, 1, 2, ..., upto $H^{-1}(\frac{\mathcal{H}}{R})n$ 1’s.
2. Since the bits in the prefixes in the MMLZ algorithm are chosen in an i.i.d. manner with probability $H^{-1}(\frac{\mathcal{H}}{R})$ of being a ‘1’, the prefixes in the MMLZ algorithm all will have $H^{-1}(\frac{\mathcal{H}}{R})n$ ones.
3. Thus a prefix in the MLZ algorithm will have equal or lesser number of 1’s than the MMLZ algorithm due to which the output of the MLZ algorithm will have equal or lesser number of 1’s (or transitions) compared to the MMLZ algorithm.

Since the MLZ algorithm will have equal or lesser number of 1’s (or transitions) than the MMLZ algorithm and the MMLZ algorithm is asymptotically optimal, the MLZ algorithm is also asymptotically optimal. ¶

REFERENCES

- [1] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*, Prentice-Hall, Englewood Cliffs, NJ, 1990.
- [2] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, “Asymptotic Zero-Transition Activity Encoding for Address Buses in Low-Power Microprocessor-Based Systems,” *Great Lakes VLSI Symp.*, pp. 77–82, Urbana IL, March 13–15 1997.
- [3] A. P. Chandrakasan and R. W. Brodersen, “Minimizing power consumption in digital CMOS circuits,” *Proceedings of the IEEE*, vol. 83, no. 4, pp. 498–523, April 1995.

- [4] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473-484, April 1992.
- [5] T. M. Cover and J. A. Thomas, "Elements of Information Theory," John Wiley & Sons, 1991.
- [6] D. W. Faulkner, "PCM Signal Coding," *U.S. Patent no. 5,062,152*, October 1991.
- [7] R. J. Fletcher, "Integrated Circuit Having Outputs Configured for Reduced State Changes," *U.S. Patent no. 4,667,337*, May 1987.
- [8] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," *IEEE Symposium on Low Power Electronics and Design*, pp. 8-11, San Diego, CA, October 10-12 1994.
- [9] S. Iman and M. Pedram, "An Approach for Multilevel Logic Optimization Targeting Low Power," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 8, pp. 889-901, August 1996.
- [10] D. Marculescu, R. Marculescu, and M. Pedram, "Information theoretic measures for power analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 6, pp. 599-610, June 1996.
- [11] R. Marculescu, D. Marculescu, and M. Pedram, "Switching Activity Analysis Considering Spatiotemporal Correlations," *IEEE/ACM International Conference on Computer-Aided Design* pp. 294-299, San Jose CA, November 6-10 1994.
- [12] M. Nemani and F. Najm, "Towards a High-Level Power Estimation Capability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 6, pp. 588-598, June 1996.
- [13] K. K. Parhi, "Algorithm transformation techniques for concurrent processors," *Proceedings of the IEEE*, vol. 77, pp. 1879-1895, December 1989.
- [14] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "Coding For Low-Power Address And Data Busses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, (to appear).
- [15] N. R. Shanbhag, "A mathematical basis for power-reduction in digital VLSI systems," *IEEE Transactions on Circuits and Systems, Part II*, vol. 44, no. 11, pp. 935-951, November 1997.
- [16] ———, "Lower bounds on Power-Dissipation for DSP algorithms," *International Symposium on Low Power Electronics and Design*, pp. 43-48, Monterey, CA, August 12-14 1996.
- [17] ———, "A fundamental basis for power-reduction in VLSI circuits," *International Symposium on Circuits and Systems*, vol. 4, pp. 9-12, Atlanta, GA, May 12-15 1996.
- [18] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, "On average power dissipation and random pattern testability of CMOS combinational logic networks," *International Conference on Computer-Aided Design*, pp. 402-407, Santa Clara, CA, November 8-12 1992.
- [19] M. R. Stan and W. P. Bursleson, "Two-dimensional Codes for Low-Power," *International Symposium on Low-Power Electronics and Design*, pp. 335-340, Monterey, CA, August 12-14 1996.
- [20] ———, "Bus-Invert Coding for Low-Power I/O," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 4, pp. 49-58, March 1995.
- [21] ———, "Coding a Terminated Bus for Low Power," *Great Lakes Symposium on VLSI*, pp. 70-73, Buffalo, NY, March 1995.
- [22] ———, "Limited-Weight Codes for Low-Power I/O," *International Workshop on Low Power Design*, pp. 209-214, April 1994.
- [23] C. L. Su, C. Y. Tsui, and A. M. Despaign, "Saving Power in the Control Path of Embedded Processors," *IEEE Design and Test of Computers*, vol. 11, no. 4, pp. 24-30, Winter 1994.
- [24] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, pp. 8-19, 1984.
- [25] J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337-343, 1977.

Sumant Ramprasad received the Bachelor of Technology (B.Tech.) degree in computer science and engineering in 1988 from the Indian Institute of Technology, Bombay, India. He received the M.S. degree in computer and information sciences from the Ohio State University, Columbus, in 1990. From 1991 to 1996 he worked at the Chicago Corporate Research Laboratories of Motorola Inc. He is a Ph.D. candidate in computer science at the University of Illinois at Urbana-Champaign.

Currently, he is working on high-level estimation, synthesis, and methodologies for low-power design.

Naresh R. Shanbhag (S'87-M'88) received the B.Tech from Indian Institute of Technology, New Delhi, India, in 1988, M.S. from Wright State University in 1990, and Ph.D. degree from University of Minnesota, in 1993, all in Electrical Engineering. From July 1993 to August 1995, he worked at AT&T Bell Laboratories at Murray Hill in the Wide-Area Networks Group, where he was responsible for development of VLSI algorithms, architectures and implementation for high-speed data communications applications. In particular, he was the lead chip architect for AT&T's 51.84 Mb/s transceiver chips over twisted-pair wiring for Asynchronous Transfer Mode (ATM)-LAN and broadband access applications. Since August 1995, he is a Research Assistant Professor with the Coordinated Science Laboratory in the VLSI Circuits Group and an Assistant Professor with the Dept. of Electrical & Computer Engg. at Univ. of Illinois at Urbana-Champaign.

His research interests are in exploring the limits of computation in an integrated circuit media in terms of power dissipation, reliability and throughput; and developing VLSI algorithms, architectures, and integrated circuits for signal processing and communication systems. He has published more than 20 journal articles/book chapters and more than 30 conference publications on these subjects.

Dr. Shanbhag received the 1999 IEEE Leon K. Kirchmayer Best Paper Award, the NSF CAREER Award in 1996, and the 1994 Darlington Best Paper Award from the IEEE Circuits and Systems society. He is a Distinguished Lecturer of the IEEE Circuits and Systems Society (97-99). Since July 1997, he is serving as an Associate Editor for IEEE Transaction on Circuits and Systems: Part II.

Dr. Shanbhag received the 1999 IEEE Leon K. Kirchmayer Best Paper Award, the NSF CAREER Award in 1996, and the 1994 Darlington Best Paper Award from the IEEE Circuits and Systems society. He is a Distinguished Lecturer of the IEEE Circuits and Systems Society (97-99). Since July 1997, he is serving as an Associate Editor for IEEE Transaction on Circuits and Systems: Part II.

Ibrahim N. Hajj (S'64-M'70-SM'82-F'90) is a Professor of Electrical & Computer Engg. and a Research Professor at the Coordinated Science Laboratory, Univ. of Illinois, Urbana-Champaign, Illinois. He received his B.E. degree (with distinction) from the American University of Beirut, the M.S. degree from the Univ. of New Mexico, Albuquerque, and the Ph.D. degree from the Univ. of California at Berkeley, all in Electrical Engineering. Before joining the University of Illinois, he was with

the Dept. of Electrical Engg. at the University of Waterloo, Ontario, Canada.

His current research interests include computer-aided design of VLSI circuits, design for reliability and low-power, synthesis, physical design, and testing. He has published over 160 journal and conference papers and book chapters on these subjects. He is a co-author of a book, "Switch-Level Timing Simulation of MOS VLSI Circuits," Kluwer Academic Publishers, 1989. In 1992 he was a co-recipient of the IEEE Transactions on Computer-Aided Design Best Paper Award.

Dr. Hajj is a Fellow of IEEE, and currently serves on the Board of Governors of the IEEE Circuits and Systems Society. He has served as an Associate Editor of the IEEE Transactions on Circuits and Systems, and an Associate Editor of the IEEE Circuits and Systems Magazine. He is a member of the Computer-Aided Network Design (CANDE), a member of ACM, and a member of Sigma Xi.