Information-Theoretic Bounds on Average Signal Transition Activity

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

Abstract—Transitions on high-capacitance busses in very large scale integration 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 ones (or zeros). These results are applied to determine the activity-reducing efficiency of different coding algorithms such as entropy coding, transition signaling, and bus-invert coding and 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 eight symbols and 13 symbols, respectively, are coded at a time.

Index Terms—Achievable bounds, busses, CMOS circuits, information theory, low power, switching activity.

I. INTRODUCTION

Power dissipation has become a critical very large scale integration (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

$$P_D = \frac{1}{2} T C_L V_{dd}^2 f$$

(1)

where $T$ is the transition activity at the node, $C_L$ is the capacitance (the product $T C_L$ 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 $C_L$, which 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 peak number of transitions. Hence, the signal encoding approaches in [2], [7], [14], [19], [20], and [23] achieve power reduction by reducing $T$ in (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 of Su et al. [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 1 b 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 ones 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 equivalence has provided lower bounds on power dissipation for digital VLSI systems [16], [17] and, 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] and [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-b 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 multibit 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 this 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 this 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]:

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

(2)

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 be identically equal to zero [from (2)].

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

$$H(X_1, X_2, \cdots, X_n) = -\sum_{x_1, x_2, \cdots, x_n} p(x_1, x_2, \cdots, x_n) \log_2 p(x_1, x_2, \cdots, x_n).$$

(3)

The entropy rate of a stochastic process $\{X_t\}$ is defined as

$$H = \lim_{n \to \infty} \frac{1}{n} H(X_1, X_2, \cdots, X_n) \text{ bits}$$

(4)

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:

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

(5)

The function $H(x)$, shown in Fig. 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 zero.
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 $(1/n) \sum_{i=1}^{n} H(x_i) \leq H((1/n) \sum_{i=1}^{n} x_i)$.
4) The derivative of $H(x)$ with respect to $x$,

$$H'(x) = \log_2 \left( \frac{1}{x} \right) \text{ bits.}$$

5) $H(x)$ is monotonically increasing in the interval $(0,1/2]$ because $H'(x) \geq 0$ in the interval with the equality occurring only at $x = 1/2$.
6) $H(x)$ is monotonically decreasing in the interval $[1/2, 1)$ because $H'(x) \leq 0$ in the interval with the equality occurring only at $x = 1/2$.

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

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

(6)

The function $H^{-1}(\cdot)$ maps the entropy of a binary-valued discrete random variable to a probability value that lies between zero and $1/2$. Following the convention in literature, we denote three different functions as $H(\cdot)$. In (2), the argument of $H(\cdot)$ is a single random variable, in (3), the argument of $H(\cdot)$ is a sequence of random variables and, in (5), the argument of $H(\cdot)$ is a real number between zero and one. 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 \( H^{-1}(y) \) given \( y \leq H(x) \). Lemma 2 employs Lemma 1 to bound the expected number of ones 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

\[
H^{-1}(y) \leq x \leq 1 - H^{-1}(y).
\]  
(7)

The relation between \( x \) and \( y \) in Lemma 1 is shown in Fig. 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 if
3) \( p_b = \lim_{n \to \infty} (1/n) \sum_{i=1}^{n} p_i \) exists then

\[
H^{-1}(\mathcal{H}) \leq p_b \leq 1 - H^{-1}(\mathcal{H})
\]  
(8)

and the bounds in (8) 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 \( B_1, B_2, \ldots \), 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., \( \lim_{n \to \infty} (1/n) \sum_{i=1}^{n} B_i \oplus B_{\text{prev}(i)} \) exists, where the function prev(\( i \)) returns the index of the bit that immediately precedes \( B_i \) on the same wire and \( \oplus \) is the exclusive-or operator), then

\[
H^{-1}\left(\frac{\mathcal{H}}{R}\right) T \leq T \leq \left(1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right)\right) R
\]  
(9)

and the bounds in (9) are asymptotically achievable.

In (9), \( H^{-1}(\mathcal{H}/R) \) is the lower bound on the bit-level transition activity (this will be proven 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 Fig. 2. Any coding algorithm (such as those in [2], [7], [14], [19], [20], and [23]) will need to reside in the region shown in Fig. 2.

The lower bound on transition activity computed by Theorem 1 for different values of \( R \) is shown on a larger scale in Fig. 3, where we see that the transition activity can be made arbitrarily close to zero 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 \( 1 - H^{-1}(\mathcal{H}/R) \), thereby achieving the upper bound in (9).

**Remark 1:** The bounds on the expected number of ones per symbol \( P \) are the same as those for \( T \), i.e.,

\[
H^{-1}\left(\frac{\mathcal{H}}{R}\right) R \leq P \leq \left(1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right)\right) R
\]  
(10)

The proof of (10) is similar to that of Theorem 1.
A. An Asymptotically Optimal Coding Algorithm

We now present a modified Lempel–Ziv (MLZ) algorithm, which 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 (9). While the MLZ algorithm will minimize $T$ in (9), 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.

Pass 1: The source sequence is sequentially parsed into phrases that have not appeared thus far. For example, if the string is $101010100010\ldots$, we parse it as $1$, $0$, $11$, $01$, $00$, $10$, $\ldots$. 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.

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 $(nR/c(n))-1$ and $1$ b to describe the last bit. We choose the addresses in order of increasing the number of ones, i.e., the all zeros 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 $(nR/c(n)) - 1$ 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.

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 $(nR/c(n)) - 1$ 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 ones 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, $T = (H/2)$.

**Proof:**

\[
H^{-1}\left(\frac{H}{R}\right)R \leq T \leq \left(1 - H^{-1}\left(\frac{H}{R}\right)\right)R \quad [9]
\]

\[
\Rightarrow H^{-1}(1)H \leq T \leq \left(1 - H^{-1}(1)\right)H
\]

[Since $R = H$ for fully compressed data]

\[
\Rightarrow T = \frac{H}{2} \quad \text{[Since } H^{-1}(1) = \frac{1}{2}\text{].}
\]

Therefore, as shown in Fig. 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 subsection: 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 $\{0, 1\}$ when $R = 1$ bit/symbol.

**Proof:**

\[
H^{-1}\left(\frac{H}{R}\right)R \leq T \leq \left(1 - H^{-1}\left(\frac{H}{R}\right)\right)R \quad [9]
\]

\[
\Rightarrow H^{-1}(H) \leq T \leq 1 - H^{-1}(H) \quad [\text{Since } R = 1].
\]

Since the source is i.i.d. with alphabet $\{0, 1\}$, $H^{-1}(H)$ 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 $1 - H^{-1}(H)$ is greater than or equal to the upper bound of $H/2$ 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 $1 - H^{-1}(H)$ for the same entropy rate if the more probable symbol is signaling with a transition.
C. Bounds for 1-b 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 b of redundancy. Thus, \( R = \mathcal{H}+1 \) and, hence, from Theorem 1, the expected transition activity is bounded by

\[
(H+1)H^{-1}\left(\frac{H}{H+1}\right) \leq T \leq (H+1)\left(1-H^{-1}\left(\frac{H}{H+1}\right)\right),
\]

(11)

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 (11) 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 (11) 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 \( (\mathcal{H}/\mathcal{H}+1) \) in (11) approaches one and \( T \) approaches \( \mathcal{H}/2 \). Thus, as the entropy rate increases, the benefit of bus-invert coding or any 1-b redundant code is reduced. The above analysis can also be extended for a \( k \)-b redundant code.

D. Lower Bound on Power-Delay Product

If the capacitance \( C_L \), supply voltage \( V_{dd} \), and 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 \( \text{Power Delay}_{\text{min}} \) given \( \mathcal{H} \) and \( R \) is given by

\[
\text{Power Delay}_{\text{min}} = KH^{-1}\left(\frac{H}{R}\right)R^2
\]

(12)

where \( K \) is a constant of proportionality. The graph of \( \text{Power Delay}_{\text{min}} \) versus \( R \) for a given value of \( \mathcal{H} \) is shown in Fig. 4. For given \( \mathcal{H} \), we can find the \( R \) that minimizes \( \text{Power Delay}_{\text{min}} \) by equating the derivative of (12) with respect to \( R \) to zero. The value of \( R \) that minimizes \( \text{Power Delay}_{\text{min}} \) is found to be

\[
R_{\text{min, power delay}} = 1.25382 \mathcal{H}.
\]

(13)

Thus, (13) 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 nonoptimal 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 five-symbol alphabet \( \mathcal{X} = \{A, B, C, D, E\} \) with probabilities \((1/2), (1/4), (1/8), (1/16), \) and \((1/16)\), respectively. Since the source is i.i.d., the entropy rate is equal to the entropy and is given by \( \mathcal{H} = (15/8) \) bits. Assume an average of \( R = 3 \) bits are employed to code a symbol. Thus, \( (\mathcal{H}/R) = (5/8) \), \( H^{-1}(\mathcal{H}/R) = 0.156142 \), and, from Theorem 1, the bounds on transition activity are

\[
0.468426 \leq T \leq 2.531574.
\]

(14)

We now calculate the actual transition activity that is achieved by various coding algorithms and compare them with the bounds in (14). 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 ones. 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.

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 \( E = 11111 \). 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 \( \mathcal{H} = (5/8) \), the 15 bits correspond to an average of eight symbols and, hence, employing 24 bits results in an expected number of three bits/symbol. 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 \( \mathcal{H} = (5/8) \), the 15 bits correspond to an average of eight symbols and, hence, employing 24 bits results in an expected number of three 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 five-limited weight code [21], [22] in which we employ the all-zero (or no transitions) code, 24 codes with a single one, 276 codes with two 1’s, etc., up to 19,817 of the codes with five 1’s. The expected number of transitions (or 1’s) in each transmission of 24 bits is given by [14]

\[
\Sigma_{i=0}^{24} \left(\frac{24}{i}\right) + 5 \Sigma_{i=0}^{24} \left(\frac{24}{i}\right) = 4.52383 \text{ transitions.}
\]

Since 24 bits correspond to an average of eight symbols, the expected number of transitions per symbol is given by

\[
(1/8) \times 4.52383 = 0.565478 \text{ transitions/symbol},
\]

which is 20.7% above the lower bound in (14).
2) Probability-Based Coding: An alternative algorithm to reduce the number of transitions is to code the most probable symbol as 000 or no transitions, as 001, as 010, as 100, and as 011. The expected number of transitions per symbol is \( T = 0.5625 \) transitions/symbol, which is 20.1% above the lower bound in (14). 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 Fig. 5. Thus, we can achieve a transition activity within 4% of the lower bound by employing a block size of eight.

**F. Bounds on Transition Activity for a Markov Process**

Consider the three-state stationary Markov process \( U_1, U_2, \ldots \), 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 three codes \( C_1, C_2, \) 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 \( C_i \) (for an example, see Table II). 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 \( \mu = \left[ (2/9), (4/9), (1/3) \right]^T \). The entropy rate of the stationary Markov process is given by [5]

\[
H(X) = -\sum_{ij} \mu_i P_{ij} \log_2 P_{ij} = \frac{4}{3} \text{ bits.} \quad (15)
\]

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

\[
0.347904 \leq T \leq 1.652096 \text{ transitions/symbol.} \quad (16)
\]

The transitions/symbol for entropy coding followed by redundancy coding and probability-based coding is shown in Fig. 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: 1) we can achieve transition activities within 4% and 9% of the lower bound with block sizes of 8 and 13 symbols, respectively (see Figs. 5 and 6); 2) transition activity can be reduced by coding larger and larger blocks of symbols (see Figs. 5 and 6); and 3) 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, nonempirical relation between the transition activity per symbol \( T \) and the entropy rate \( H \) of a process by deriving lower and upper bounds on the expected transition activity per symbol and the expected number of ones per symbol given the entropy rate \( H \) 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 \( H \) and \( R \).

In the 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 ≤ 1) and \( y = 0 \) as follows.

1) Case 1: \( 0 < y \leq 1 \): Thus, \( 0 < H^{-1}(y) \leq (1/2) \). There are three cases for \( x \):
   - \( x = 1 \) or \( x = 0 \): This is not possible because \( H(x) = 0 \) and \( y > 0 \), which would violate \( y \leq H(x) \).
   - \( 1/2 \leq x < 1 \):
     \[
     H(1 - H^{-1}(y)) \leq H(x)
     \]
     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 \( \left[ \frac{1}{2}, 1 \right] \)
     \[ \Rightarrow H^{-1}(y) \leq x \leq 1 - H^{-1}(y) \]
     Since \( 0 < H^{-1}(y) \leq \frac{1}{2} \) and \( \frac{1}{2} \leq x < 1 \).

2) Case 2: \( y = 0 \): Since \( H^{-1}(y) = 0 \) and \( 0 \leq x \leq 1 \), (7) is satisfied.

Hence, the proof.

APPENDIX B

PROOF OF LEMMA 2

From the definitions of entropy rate in (4) and \( \mathcal{H} \) in the statement of Lemma 2,

\[
\mathcal{H} \leq \lim_{n \to \infty} \frac{1}{n} H(B_1, B_2, \ldots, B_n) \leq \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} H(B_i)
\]

[Independence bound on entropy]

\[ = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} H(p_i) \bar{p} = \Pr(B_i = 1) \]

\[ \leq H \left( \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} p_i \right) \]

[Property 3: Jensen’s inequality and concavity of \( H \)]

\[ \Rightarrow \mathcal{H} \leq H(p_b) \]

[By definition, \( p_b = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} p_i \).]

Thus, we can substitute \( \mathcal{H} \) for \( y \) and \( p_b \) for \( x \) in Lemma 1 to obtain (8).

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 two symbols. The asymptotic equipartition property (AEP) [5] states that given a stationary and ergodic process, for each \( \epsilon_1 > 0 \), there exists \( n_1 \) suchthat for all \( n > n_1 \) the following properties hold:
   - There is a set, called the typical set \( A_{n_1}^n \), which is a subset of the set of all possible sequences of \( n \) symbols generated by the process.
   - The number of elements in \( A_{n_1}^n \), \( |A_{n_1}^n| \), is bounded by
     \[
     (1 - \epsilon_1)2^{n(H(\cdot) - \epsilon_1)} \leq |A_{n_1}^n| \leq 2^{n(H(\cdot) + \epsilon_1)}. \tag{17}
     \]
   - The probability of \( A_{n_1}^n \) containing a sequence of \( n \) symbols generated by the process is at least \( 1 - \epsilon_1 \).
   - \( \epsilon_1 \to 0 \) as \( n \to \infty \).

   In short, AEP states that as the length of the sequence \( n \) increases, the probability that a generated sequence belongs to \( A_{n_1}^n \) approaches unity and the size of the typical set \( |A_{n_1}^n| \) approaches \( 2^{n \mathcal{H}} \).

2) We generate a set \( C_n^\infty \) of codewords. Each codeword in \( C_n^\infty \) is formed by drawing \( n \) code bits in an i.i.d. manner with probability \( p \) of being a “1.” Again, from AEP, we know that the set \( C_n^\infty \) will contain at least \( (1 - \epsilon_2)2^{n(H(\cdot) - \epsilon_2)} \) distinct codewords. We choose \( p \) such that the number of codewords in \( C_n^\infty \) is at least the number of sequences in \( A_{n_1}^n \), i.e.,

\[
(1 - \epsilon_2)2^{n(H(\cdot) - \epsilon_2)} = 2^{n(H(\cdot) + \epsilon_1)}.
\]

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

As \( n \to \infty \), both \( \epsilon_1 \) and \( \epsilon_2 \to 0 \) and \( p \to H^{-1}(\mathcal{H}) \).

3) We assign codewords to sequences in \( A_{n_1}^n \) from the set \( C_n^\infty \).

4) After each sequence in \( A_{n_1}^n \) has been assigned a codeword from \( C_n^\infty \), sequences not in \( A_{n_1}^n \) are assigned codewords in an arbitrary manner from the remaining codewords (which may or may not be in \( C_n^\infty \)).

As \( n \to \infty \), the probability of a “1” at the output of the L2 encoder is at most \( (1 - \epsilon_1)p_n + \epsilon_1 n/n \), where: 1) the probability of a sequence being in \( A_{n_1}^n \) is at least \( 1 - \epsilon_1 \) (AEP); 2) as \( n \to \infty \), the number of ones in a codeword encoding a sequence from \( A_{n_1}^n \) is \( p_n \) (strong law of large numbers); 3) the probability of a sequence not being in \( A_{n_1}^n \) is at most \( \epsilon_1 \) (AEP); and 4) the number of ones in a codeword encoding a sequence not from \( A_{n_1}^n \) is at most \( n \) (since length of codeword is \( n \)).

Hence, as \( n \to \infty \), 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 ones and zeros. \( \blacksquare \)
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 one represents a transition and a zero represents no transition. We will then employ Lemma 2 to bound the number of ones 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 $(X_1, X_2, \ldots, X_N)$, generated by the process. The symbols are coded, independently of any other symbols, to the $n = NR$ bits represented by the binary random variables $B_1, B_2, \ldots, B_n$ and transmits these $n$ bits in any order. Clearly, as $N$ (or $n$) $\to \infty$, 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 $n \to \infty$ 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.,

$$ (X_1, X_2, \ldots, X_N) = f(B_1, B_2, \ldots, B_n). \quad (18) $$

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

$$ \lim_{n \to \infty} \frac{1}{n} H(B_1, B_2, \ldots, B_n) \geq \lim_{n \to \infty} \frac{1}{n} H(f(B_1, B_2, \ldots, B_n)) $$

[Joint entropy of a collection of random variables

$$ \geq \text{Joint entropy of a function of the collection of random variables} $$

$$ = \lim_{N \to \infty} \frac{1}{NR} H(X_1, X_2, \ldots, X_N) $$

$$ = \frac{H}{R} $$

$$ \Rightarrow \lim_{n \to \infty} \frac{1}{n} H(B_1, B_2, \ldots, B_n) \geq \frac{H}{R}. \quad (19) $$

Define a function $g$ on $(B_1, B_2, \ldots, B_n)$ as follows:

$$ (C_1, C_2, \ldots, C_n) = g(B_1, B_2, \ldots, B_n) \quad (20) $$

where

$$ C_i = B_i \oplus B_{\text{prev}(i)} \quad (21) $$

where the function $\text{prev}(i)$ 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 $B_{\text{prev}(i)}$ is "0." Hence, we can recursively compute $(B_1, B_2, \ldots, B_n)$ given $(C_1, C_2, \ldots, C_n)$ as follows:

$$ B_i = C_i \oplus B_{\text{prev}(i)} \quad (22) $$

Fig. 7 shows the relation between $B_i$, $B_{\text{prev}(i)}$, and $C_i$. The two delays in Fig. 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 $B_{\text{prev}(i)}$ 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 $B_{\text{prev}(j)}$. Since $g$ is an invertible function

$$ \lim_{n \to \infty} \frac{1}{n} H(C_1, C_2, \ldots, C_n) = \lim_{n \to \infty} \frac{1}{n} H(B_1, B_2, \ldots, B_n), \quad (23) $$

From (19) and (23), we get

$$ \lim_{n \to \infty} \frac{1}{n} H(C_1, C_2, \ldots, C_n) \geq \frac{H}{R}. \quad (24) $$

Let $p_c = \lim_{n \to \infty} (1/n) \sum_{C_j} p_{C_j}$, 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, $p_{\text{on}}$ is the number of ones in $(C_1, C_2, \ldots, C_n)$ for large $n$. Substituting $p_c$ for $p_{\text{on}}$ and $\mathcal{H}/R$ for $\mathcal{H}$ in Lemma 2, we obtain

$$ H^{-1} \left( \frac{\mathcal{H}}{R} \right) \leq p_c \leq 1 - H^{-1} \left( \frac{\mathcal{H}}{R} \right). \quad (25) $$

Since there are on the average $R$ bits per symbol and $C_i$ is one if and only if there was a transition at $B_i$, we have

$$ T = p_c R. \quad (26) $$

Multiplying (25) by $R$ and employing (26), we have

$$ H^{-1} \left( \frac{\mathcal{H}}{R} \right) R \leq T \leq \left( 1 - H^{-1} \left( \frac{\mathcal{H}}{R} \right) \right) R \quad (27) $$

which is the desired result.

Proof of Asymptotic Achievability of Theorem 1: We now present a coding algorithm $T_1$, which asymptotically achieves the lower bound on transition activity in Theorem 1 for stationary and ergodic processes. The $T_1$ coding algorithm is similar to the $L_2$ algorithm with the differences being that the source is not binary and $R$ is now not restricted to being one. We provide an outline of the proof of asymptotic optimality of the $T_1$ algorithm. The detailed proof is similar to the proof of asymptotic optimality of the $L_2$ algorithm.

1) We first group blocks of $k$ symbols from the source. For large $k$, there are $2^k R$ 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 $L_2$ algorithm, each bit in a codeword encoding a block of symbols is chosen in an i.i.d. manner with probability $H^{-1}(\mathcal{H}/R)$ of being one. Thus, there are $2^k R H^{-1}(\mathcal{H}/R) = 2^k R$ codewords (AEP). Hence, we have a codeword for each block of symbols. Since each

![Fig. 7. Relation between $B_i$, $B_{\text{prev}(i)}$, and $C_i$.](image)
bit in each codeword has probability $H^{-1}(H/R)$ of being one, the bit-level probability at the output of the T1 encoder is also $H^{-1}(H/R)$.

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 $H^{-1}(H/R)$, which is then scaled by $R$ to achieve the lower bound on transitions per symbol.

\section*{Appendix D}

\textbf{Proof of Asymptotic Optimality of the MLZ Algorithm}

Outline of Proof: We first prove the asymptotic optimality of another algorithm, the modified MLZ 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 MMLZ is identical to the MLZ algorithm, except for pass 2 in Section III-A, where the bits in the prefix are chosen, not in order of increasing the number of ones, but in an i.i.d. manner with probability $p$ of being a one.

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

$$c(n) = (1 - c)p^{2(nR/c(n)+1)(H(p))^{-1}}.$$ \vspace{1ex}

As $n \to \infty$, $c(n) \to 0$ and from the proof of optimality of the original Lempel–Ziv algorithm [5], we know that

$$\lim_{n \to \infty} \frac{c(n) \log_2 c(n)}{n} = H,$$ \hspace{1ex} (28)

$$\lim_{n \to \infty} \frac{c(n)}{n} = 0.$$ \hspace{1ex} (29)

Hence,

$$\lim_{n \to \infty} p = H^{-1}\left(\frac{H}{R}\right).$$ \hspace{1ex} (30)

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

$$p_{\text{MMLZ}} \leq \frac{c(n)\left((nR/c(n) - 1)p + 1\right)}{nR}.$$ \hspace{1ex} (31)

We can obtain (31) as follows.

1) As $n \to \infty$, the number of ones in the prefix is $((nR/c(n) - 1)p + 1)$ (strong law of large numbers).

2) $((nR/c(n) - 1)p + 1)$ is the maximum number of ones, including the prefix and additional bit.

3) $c(n)\left((nR/c(n) - 1)p + 1\right)$ is the maximum number of ones in the coded sequence, which is then divided by the total number of bits $nR$ to give the upper bound on $p_{\text{MMLZ}}$.

Simplifying (31), we obtain

$$p_{\text{MMLZ}} \leq \left(1 - \frac{c(n)}{nR}\right)p + \frac{c(n)}{nR}$$ \hspace{1ex} (32)

which approaches $p$ or $H^{-1}(H/R)$ as $n \to \infty$ [see (29)]. Thus by selecting the bits in the prefix in an i.i.d. manner with probability $H^{-1}(H/R)$, we arrive at an asymptotically optimal algorithm. The MLZ algorithm in Section III-A, which selects the prefixes in order of increasing the number of ones, will always be better than or equal to the MMLZ algorithm, in which all bits in the prefix have probability $H^{-1}(H/R)$ of being a one. The reason for this is as follows.

1) Since the bits in the prefix are chosen in order of increasing number of ones in the MLZ algorithm, the prefixes in the MLZ algorithm will have 0, 1, 2, ..., up to $H^{-1}(H/R)n$ ones.

2) Since the bits in the prefixes in the MMLZ algorithm are chosen in an i.i.d. manner with probability $H^{-1}(H/R)$ of being a one, the prefixes in the MMLZ algorithm will all have $H^{-1}(H/R)n$ ones.

3) Thus, a prefix in the MLZ algorithm will have equal or lesser number of ones than the MMLZ algorithm due to which output of the MLZ algorithm will have equal or lesser number of ones (or transitions) compared to the MMLZ algorithm.

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

\section*{References}


### Sumant Ramprasad

Sumant Ramprasad received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Bombay, India, in 1988, the M.S. degree in computer and information sciences from Ohio State University, Columbus, in 1990, and is currently working toward the Ph.D. degree in computer science at the University of Illinois at Urbana-Champaign.

From 1991 to 1996, he was with the Chicago Corporate Research Laboratories, Motorola Inc. He is currently working on high-level estimation, synthesis, and methodologies for low-power design.

### Naresh R. Shanbhag

Naresh R. Shanbhag (S’87–M’88) received the B.Tech. degree from the Indian Institute of Technology, New Delhi, India, in 1988, the M.S. degree from Wright State University in 1990, and the Ph.D. degree from the University of Minnesota, Minneapolis, in 1993, all in electrical engineering.

From July 1993 to August 1995, he was with AT&T Bell Laboratories, Murray Hill, NJ, 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)–local area network (LAN) and broad-band access applications. Since August 1995, he has been a Research Assistant Professor with the Coordinated Science Laboratory, VLSI Circuits Group, and an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. His research interests include 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 over 20 journal articles/book chapters and over 30 conference publications on these subjects.

Dr. Shanbhag received the 1999 IEEE Leon K. Kirchmayer Best Paper Award, the 1999 Xerox Faculty Award, the 1996 NSF CAREER Award, and the 1994 Darlington Best Paper Award presented by the IEEE Circuits and Systems Society. He has served as an associate editor for the IEEE *Transactions on Circuits and Systems Part II: Analog and Digital Signal Processing* since July 1997. He was a Distinguished Lecturer of the IEEE Circuits and Systems Society from 1997 to 1999.

### Ibrahim N. Hajj

Ibrahim N. Hajj (S’64–M’70–SM’82–F’90) received the B.E. degree (with distinction) from the American University of Beirut, Beirut, Lebanon, the M.S. degree from the University of New Mexico, Albuquerque, and the Ph.D. degree from the University of California at Berkeley, all in electrical engineering.

He is currently a Professor of electrical and computer engineering and a Research Professor at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. Prior to joining the University of Illinois at Urbana-Champaign, he was with the Department of Electrical Engineering at the University of Waterloo, Waterloo, Ont., 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 co-authored *Switch-Level Timing Simulation of MOS VLSI Circuits* (Norwell, MA: Kluwer, 1989).

Dr. Hajj is a member of the Computer-Aided Network Design (CANDE), Association for Computing Machinery (ACM), and Sigma Xi. In 1992, he was a co-recipient of the *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* Best Paper Award. He 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 Part II: Analog and Digital Signal Processing*, and an associate editor of the *IEEE Circuits And Systems Magazine*. 