Signal Coding for Low Power: Fundamental Limits and Practical Realizations

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

Abstract—Transitions on high-capacitance buses 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 present: 1) fundamental bounds on the activity-reduction capability of any encoding scheme for a given source and 2) practical novel encoding schemes that approach these bounds. The fundamental bounds in 1) are obtained via an information-theoretic approach, where a signal $x(n)$ with entropy rate $\mathcal{H}$ is coded with $R$ bits per sample on average. The encoding schemes in 2) are developed via a communication-theoretic approach, whereby a data source is passed through a decorrelating function followed by a variant of entropy coding function which reduces the transition activity. Simulation results with an encoding scheme for data busses indicate an average reduction in transition activity of 36%.

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

I. INTRODUCTION

POWER dissipation has become a critical VLSI design concern in recent years [2], and a substantial amount of research is being conducted 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 $P_D = (1/2)TC_LV_{dd}^2f$, where $T$ is the transition activity at the node, $C_L$ is the 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 [1], [4], [11], [12] 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 literature achieve power reduction by reducing $T$ while keeping $C_L$ more or less unaltered.

In this paper, we present fundamental coding-scheme-independent lower and upper bounds [9] on the expected transition activity obtained via an information-theoretic approach, where a signal $x(n)$ with entropy rate $\mathcal{H}$ is coded with $R$ bits per sample on average. We then present a communication-theoretic, source-coding framework for the design of coding schemes that approach these bounds [10]. These schemes are suited for high-capacitance busses, where the extra power dissipation due to the encoder and the decoder circuitry is offset by the power savings in the bus. In the proposed source coding framework, a data source (characterized in a probabilistic manner) is passed through a decorrelating function $f_1$ first. Next, a variant of entropy coding function $f_2$ is employed, which reduces the transition activity. The framework is then employed to derive novel encoding schemes whereby practical forms for $f_1$ and $f_2$ are proposed. Simulation results with an encoding scheme for data busses indicate an average reduction in transition activity of 36%. We then examine the transition activity reducing efficiency of these coding schemes. This work is a continuation of our effort in developing an information-theoretic view of VLSI computation [8], whereby equivalence between computation and communication is being established.

The concept of entropy from information theory was employed in the area of high-level power estimation in [6], [7]. In [7], 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 [6], information theory is employed to estimate power dissipation at logic and register-transfer levels. The work presented here is applicable to multi-bit signals, is 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.

The rest of this paper is organized as follows. In Section II, we present the bounds on transition activity and in Section III, we present practical coding schemes to approach these bounds.

II. INFORMATION-THEORETIC BOUNDS ON TRANSITION ACTIVITY

In this section, we present achievable lower and upper bounds on the expected number of transitions and then provide applications. We first 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 $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$ bits [3]. The joint entropy $H(X_1, X_2, \ldots, X_n)$ of a collection
of discrete random variables \((X_1, X_2, \ldots, X_n)\) with a joint distribution \(p(x_1, x_2, \ldots, x_n)\) is defined as
\[
H(X_1, X_2, \ldots, X_n) = -\sum p(x_1, x_2, \ldots, x_n) \log_2 p(x_1, x_2, \ldots, x_n).
\] (2.1)

The entropy rate of a stochastic process \(\{X_t\}\) is defined as
\[
\mathcal{H} = \lim_{n \to \infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) \text{ bits}
\] (2.2)
when the limit exists. For an independent and 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 \(H(x) = -x \log_2 x - (1-x) \log_2 (1-x)\). 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 \(x\), \(H'(x) = \log_2(1-x/x)\).
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 \(H\) is defined on the real interval \([0, 1]\) as \(H^{-1}(y) = y\) if \(H(y) = x\) and \(y \in [0, (1/2)]\).

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/symbol of a process with a certain entropy rate, given that each symbol is coded employing an expected number of \(R\) bits. Theorem 1 is valid whether the bits are sent serially or in parallel. The proofs are presented in the Appendixes A, B, and C.

\textbf{Lemma 1:} For all \((x, y)\) such that \(x \in [0, 1]\) and \(y \in [0, 1]\), if \(y \leq H(x)\) then
\[
H^{-1}(y) \leq x \leq 1 - H^{-1}(y).
\] (2.3)

The relation between \(x\) and \(y\) in Lemma 1 is shown in Fig. 1, where (2.3) can be seen to be true.

\textbf{Lemma 2:} If:
1) \(\{B_i\}\) is a 0–1 valued random process with entropy rate greater than or equal to \(\mathcal{H}\);
2) \(p_i\) is the probability that the random variable \(B_i\) takes the value 1;
3) \(p_b = \lim_{n \to \infty} (1/n) \sum_{i=1}^n p_i\) exists (\(p_b\) is the mean of \(p_i\’s\)) then,
\[
H^{-1}(\mathcal{H}) \leq p_b \leq 1 - H^{-1}(\mathcal{H})
\] (2.4)
and the bounds in (2.4) are asymptotically achievable if \(\{B_i\}\) is a stationary and ergodic process.

\textbf{Theorem 1:} Let:
1) \(\mathcal{H}\) be the entropy rate of a process \(\{X_t\}\);
2) the symbols be coded in a uniquely decodable manner into bits \(\{B_i\}\) 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;
4) \(T\) be the expected number of transitions in the bits on the wires/symbol (i.e., \(\lim_{n \to \infty} (1/n) \sum_{i=1}^n B_{\text{prev}(i)} \oplus B_{\text{prev}(i)}\) 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).

Then
\[
H^{-1}\left(\frac{\mathcal{H}}{R}\right) R \leq T \leq \left(1 - H^{-1}\left(\frac{\mathcal{H}}{R}\right)\right) R
\] (2.5)
and the bounds in (2.5) are asymptotically achievable if \(\{X_t\}\) is a stationary and ergodic process.

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 will need to reside in the region shown in Fig. 2. In Theorem 1, we are assuming that there is zero probability of error during transmission. The bounds in Theorem 1 have been extended in [5] to account for noise in transmission.

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 $R = 10 \mathcal{H}$.

**A. Applications of Bounds on Transition Activity**

In this subsection, we present three applications of the bounds presented in Theorem 1.

1) Transition Signaling: Consider a source with alphabet comprising of two symbols, “0” and “1.” We will define transition signaling as follows: we will signal the less probable symbol (“0” or “1”) with a transition and the other symbol with no transition.

**Corollary 1:** 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:**

$$H^{-1} \left( \frac{\mathcal{H}}{R} \right) R \leq T \leq \left( 1 - H^{-1} \left( \frac{\mathcal{H}}{R} \right) \right) R [(2.5)]$$

$$\Rightarrow H^{-1} \left( \mathcal{H} \right) R \leq T \leq 1 - H^{-1} \left( \mathcal{H} \right) \text{ [Since } R = 1 \text{].}$$

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 no transition.

2) 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. The delay (for instance, for transmitting the data on a bus) is proportional to $R$. The power dissipation can be reduced arbitrarily by increasing $R$, but this is not practical. The power-delay product is a measure that is used to determine a delay that will reduce power dissipation and is still practical.

The lower bound on the power-delay product at a fixed-supply voltage $\text{PowerDelay}_{\min}$, given $\mathcal{H}$ and $R$, is given by

$$\text{PowerDelay}_{\min} = K H^{-1} \left( \frac{\mathcal{H}}{R} \right) R^2 \quad (2.6)$$

where $K$ is a constant of proportionality. The graph of $\text{PowerDelay}_{\min}$ versus $R$ for a given value of $\mathcal{H}$ is shown in Fig. 3.

For given $\mathcal{H}$, the $R$ that minimizes $\text{PowerDelay}_{\min}$ is given by $R_{\min} = \frac{1.25392}{\mathcal{H}}$. Thus, a source with entropy rate $\mathcal{H}$ requires approximately an average of $1.25\mathcal{H}$ bits/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.

3) 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 $\mathcal{H} = 13/8$ 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

$$0.468426 \leq T \leq 2.531574. \quad (2.7)$$

As shown in this section, information theory provides bounds on transition activity. It does not, however, provide efficient methods to achieve these bounds. Hence, in the next section, we employ ideas from source coding to develop a framework for generating efficient coding schemes.

**III. SOURCE CODING FRAMEWORK**

In this section, we present a framework for generating efficient coding schemes to reduce transition activity. From information theory, one way to approach the bounds in the previous section is to group symbols into blocks and then code these blocks. As the blocks get larger, the transition activity approaches the bounds for certain coding schemes. The coder hardware, however, increases exponentially with the size of the block. Hence, we employ ideas from source coding to design efficient low-cost coding schemes.

The proposed source-coding framework in Fig. 4 is based upon a typical source-coder architecture. The function $f_1$ decorrelates the bit input $x(n)$. Therefore, the prediction error $e(n)$ is a function $f_2$ of the current value of $x(n)$ and the prediction $\hat{x}(n)$. The function $f_1$ could be a linear or a nonlinear function of its arguments. The prediction $\hat{x}(n)$ is a function $F(x(n-1), x(n-2), \ldots, x(n-M+1))$ of the past values of $x(n)$. For complexity reasons, we restrict ourselves to a value of $M = 2$.

The function $f_2$ employs a variant of entropy coding whereby, instead of minimizing the average number of bits at the output, it reduces the average number of transitions. The function $f_2$ employs the error $e(n)$ to generate an output $y(n)$, which has a “1” to indicate a transition, and a “0” to indicate no transition. This codeword is then passed through an XOR gate to generate the corresponding signal waveforms on the bus. The decoder employs the reverse operation to decode the data.

We now propose practical low-complexity choices for $F$, $f_1$, and $f_2$, and then evaluate the performance of encoding.
schemes employing different combinations of $F$, $f_1$, and $f_2$ in Section III-C.

A. Alternatives for $F$, $f_1$, and $f_2$

In this paper, two alternatives for $F$, referred to as Identity and Increment, are considered. The output of the Identity function is equal to its input, whereas the output of the Increment function is equal to its input plus one. The Identity function requires no hardware to implement and is useful if the data source has significant correlation.

Two alternatives for $f_1$, referred to as EXCLUSIVE-OR (xor) and difference-based mapping (dbm), are considered in this paper. The function xor is given by a bit-wise EXCLUSIVE-OR of the current input and the prediction. The difference-based mapping, dbm, returns the difference between $x(n)$ and $\hat{x}(n)$ properly adjusted, so that the output fits in the available $B$ bits. Both xor and dbm skew the original distribution for most data and, hence, enable $f_2$ to reduce the number of transitions even more.

Three possible choices for $f_2$ are considered in this paper, namely, invert (inv), probability-based mapping (pbm), and value-based mapping (vbm). In the inv function, if the number of 1’s in $x(n)$ exceeds half the number of bus lines, then the input is inverted and the inversion is signaled by setting an extra bit to “1,” else the input is not inverted and the extra bit is set to “0.” The function inv has been employed in bus-invert coding [11]. In the pbm function, the number of 1’s in the input is reduced by assigning, as in [4], codewords with fewer 1’s to the more frequently occurring input samples. We then map a “1” to a transition waveform and a “0” to a transitionless waveform employing, an EXCLUSIVE-OR gate.

After xor and dbm are applied, smaller values are generally more probable than larger values. This is especially true if $f_1$ is dbm. We employ this feature in vbm, in which codewords with fewer 1’s are assigned to smaller values and then map a “1” to a transition waveform and a “0” to a transitionless waveform employing an EXCLUSIVE-OR gate. The function vbm assumes that smaller values are more probable than larger values. The advantage of vbm over pbm is that a representative data sequence is not needed. The reduction in transitions with vbm, however, is usually lower than pbm.

B. Encoding Schemes

The proposed encoding schemes are summarized in Table I, where we have seven encoding schemes employing different combinations of $F$, $f_1$, and $f_2$. The xor-pbm scheme reduces the number of transitions by assigning fewer transitions to the more frequently occurring set of transitions in the original signal. The closest approach in literature to xor-pbm is [4], where signal samples having higher probability of occurrence are assigned codewords with fewer ON bits. In VLSI circuits, power dissipation depends on the number of transitions occurring at the capacitive nodes of the circuit. The xor-pbm scheme differs from [4] in two respects. It reduces the power dissipation by reducing the number of transitions by assigning fewer transitions to the more frequently occurring set of transitions. The xor-pbm scheme also achieves a greater reduction in transitions by skewing the input probability distribution by employing xor for $f_1$. The xor-vbm scheme has the advantage over xor-pbm of being an input-independent mapping and requiring lesser hardware to implement at the cost of typically lesser reduction in transitions. The scheme dbm-pbm requires more hardware than xor-pbm, but also reduces transitions more because the function dbm skews the input probability distribution more than xor.

The framework in Fig. 4 can be employed to derive and improve existing coding schemes. For example, the Gray coding scheme can be derived by letting $f_1(x(n), \hat{x}(n)) = x(n)$, $f_2$ be the Gray coding scheme, and removing the EXCLUSIVE-OR at the output of the encoder. The bus-invert [11] coding scheme can be obtained by letting $f_1$ be xor and $f_2$ be inv. A variant of the bus-invert coding scheme can be obtained by employing dbm instead of xor for $f_1$, resulting in the dbm-inv scheme. The scheme in [4] can be derived by letting $f_1(x(n), \hat{x}(n)) = x(n)$, $f_2$ be pbm, and removing the EXCLUSIVE-OR at the output of the encoder. An improved version of the T0 scheme in [1] can be derived from our framework by employing the Increment function for the predictor $F$ (with overflow being ignored), xor for $f_1$, and Identity function for $f_2$. This improved scheme, called inc-xor, is shown in Fig. 5. Unlike the T0 scheme, the inc-xor scheme does not require an extra bit, has a shorter critical path, and as simulation results in [10] show, provides 1.5 times more reduction in transitions for instruction address busses.

The performance of the above coding schemes with the data sets in Table II is shown in Table III. The xor-vbm scheme, as expected, results in a slightly lesser reduction in transitions than xor-pbm. We see that pbm, optimized for one video/audio sequence, performs well for other video/audio sequences, thus indicating the robustness of these schemes to variations in signal statistics. There is little change in transitions for uniformly distributed uncorrelated data (R1 data).

C. Comparison of Encoding Schemes

We now examine how close the coding schemes in Table I come to the lower bound in Theorem 1.
TABLE II

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A3</td>
<td>2.88 MB of 16 bit PCM audio data (pop music)</td>
</tr>
<tr>
<td>A4</td>
<td>2.88 MB of 16 bit PCM audio data (pop music)</td>
</tr>
<tr>
<td>A7</td>
<td>2.88 MB of 16 bit PCM audio data (classical music)</td>
</tr>
<tr>
<td>CO</td>
<td>0.80 MB of 16 bit communications channel data</td>
</tr>
<tr>
<td>V1</td>
<td>3.80 MB of 8 bit video data (miss america)</td>
</tr>
<tr>
<td>V2</td>
<td>22.7 MB of 8 bit video data (football)</td>
</tr>
<tr>
<td>V3</td>
<td>9.70 MB (380 CIF frames) of 8 bit video data (car phones)</td>
</tr>
<tr>
<td>R1</td>
<td>0.10 MB of white, uniformly distributed data</td>
</tr>
<tr>
<td>PS</td>
<td>0.10 MB Postscript file</td>
</tr>
</tbody>
</table>

TABLE III

<table>
<thead>
<tr>
<th>Data</th>
<th>x-or-pbm</th>
<th>x-or-pbm</th>
<th>dbm-pbm</th>
<th>dbm-pbm</th>
<th>x-or-pbm</th>
<th>x-or-pbm</th>
<th>opt. for</th>
<th>Redn. scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>A3</td>
<td>32</td>
<td>24</td>
<td>36</td>
<td>29</td>
<td>A3</td>
<td>32</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>38</td>
<td>29</td>
<td>41</td>
<td>30</td>
<td>A3</td>
<td>37</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td>41</td>
<td>32</td>
<td>45</td>
<td>25</td>
<td>A3</td>
<td>39</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>CO</td>
<td>26</td>
<td>4</td>
<td>28</td>
<td>28</td>
<td>A3</td>
<td>20</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>V1</td>
<td>39</td>
<td>34</td>
<td>44</td>
<td>44</td>
<td>V2</td>
<td>38</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>V2</td>
<td>33</td>
<td>26</td>
<td>39</td>
<td>39</td>
<td>V2</td>
<td>33</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>V3</td>
<td>33</td>
<td>25</td>
<td>40</td>
<td>39</td>
<td>V2</td>
<td>33</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>V2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td>43</td>
<td>22</td>
<td>38</td>
<td>25</td>
<td>PS</td>
<td>43</td>
<td>42</td>
<td></td>
</tr>
</tbody>
</table>

Assume that uniformly distributed independent data is transmitted over a \( B \)-bit bus. Hence, \( \mathcal{H} = B \). In schemes such as bus-invert coding [11], the transition activity on the bus is reduced by employing an additional bit. We now calculate the lower bound for any coding scheme that uses one bit of redundancy. Thus, \( R = \mathcal{H} + 1 = B + 1 \) and hence, from Theorem 1, the expected transition activity is bounded by

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

If \( B = 8 \) then bus-invert achieves a transition activity of 3.273 transitions/8 bits. The lower bound from Theorem 1 is 2.7569 transitions/8 bits, which can be approached by coding larger and larger blocks of bits. Now assume the bus width \( B \) is increased and the source entropy is also increased to be equal to \( B \). The ratio \( B/B+1 \) approaches 1 and \( T \) approaches \( B/2 \). Thus, as the bus is made wider, the benefit of bus-invert coding or any 1-bit redundant code is reduced for uniformly distributed independent data. The above analysis can also be extended for a \( k \)-bit redundant code.

1) Probability Based Coding: We use the i.i.d. source in Section II-A.3 to determine the transition activity for probability-based coding, in which we reduce the number of transitions by coding the most probable symbol \( A \) as 000 or no transitions, \( B = 001, C = 010, D = 100, \) and \( E = 011 \). The expected number of transitions/symbol can be calculated as \( T = 0.5025 \) transitions/symbol, which is within 21\% of the lower bound in (2.7). We can further reduce the number of transitions by applying probability based coding to a block of two symbols. The expected number of transitions/symbol can be calculated to be 0.521484 transitions/symbol. This can be reduced further by coding with block sizes larger than two. We can achieve a transition activity within 4\% of the lower bound by employing a block size of eight.

IV. CONCLUSION

In this paper, we have presented lower and upper bounds on the expected transition activity/symbol \( T \) given the entropy rate \( \mathcal{H} \) of a process and the expected number of bits \( R \) employed to code a symbol. 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 \( R \). We then presented a source-coding framework to describe encoding schemes to reduce transition activity and employed this framework to develop novel encoding schemes. The encoding schemes are suited for high-capacitance busses, where the extra capacitance due to the encoder and the decoder circuitry is offset by the savings in power dissipation at the bus. Simulation results show that two of the encoding schemes derived from the proposed framework (\( x-or-pbm \) for data busses and \( inc-xor \) for address busses) perform better than existing ones. The proposed framework allows one to develop novel low-power encoding schemes and characterize existing schemes.

APPENDIX A

PROOF OF LEMMA 1

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

1) \( x = 1 \) or \( x = 0 \): This is not possible because \( H(x) = 0 \) and \( y > 0 \), which would violate \( y \leq H(x) \).

2) \( 1/2 \leq x < 1 \):

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

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

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

[Property 6: \( H(\cdot) \) is monotonically decreasing in \( [1/2, 1] \)]

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

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

3) \( 0 < x < 1/2 \):

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

[Since \( y \leq H(x) \& H(\cdot) \) is increasing in \( (0, 1/2] \)]

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

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

Case 2: \( y = 0 \) : Since \( H^{-1}(y) = 0 \) and \( 0 \leq x \leq 1 \), (2.3) is satisfied.

Hence, the proof.
Appendix B
Proof of Lemma 2
From the definitions of entropy rate in (2.2) and \( H \) in the statement of Lemma 2,

\[
H \leq \lim_{n \to \infty} \frac{1}{n} H(B_1, B_2, \ldots, B_n),
\]

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

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

\[
\leq H(\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} p_i) \quad \text{[Property 3: Jensen’s inequality and concavity of } H]\n\]

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

Thus, we can substitute \( H \) for \( y \) and \( p_b \) for \( x \) in Lemma 1 to obtain (2.4).

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) [3] 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 \( A_{n_1}^{B_1} \), which is a subset of the set of all possible sequences of \( n \) symbols generated by the process;
   b) the number of elements in \( A_{n_1}^{B_1} \), \( |A_{n_1}^{B_1}| \), is bounded by \( (1 - \epsilon_1)2^{n(H - \epsilon_1)} \leq |A_{n_1}^{B_1}| \leq 2^{n(H + \epsilon_1)} \);
   c) the probability of \( A_{n_1}^{B_1} \) containing a sequence of \( n \) symbols generated by the process is at least \( 1 - \epsilon_1 \);
   d) \( \epsilon_1 \to 0 \) 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 \( A_{n_1}^{B_1} \) approaches unity, and the size of the typical set \( |A_{n_1}^{B_1}| \) approaches \( 2^{nH} \).

2) We generate a set, \( C_{n_0}^{B_0} \) of codewords. Each codeword in \( C_{n_0}^{B_0} \) is formed by drawing \( n_0 \) code bits in an i.i.d. manner with probability \( p \) of being a “1.” Again, from AEP, we know that the set \( C_{n_0}^{B_0} \) 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 \( C_{n_0}^{B_0} \) is at least the number of sequences in \( A_{n_1}^{B_1} \), i.e.,

\[
(1 - \epsilon_2)2^{n(H(p) - \epsilon_2)} = 2^{n(H(p) - \epsilon_2)} = 2^{n(0 + \epsilon_1)}.
\]

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

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

3) We assign codewords to sequences in \( A_{n_1}^{B_1} \) from the set \( C_{n_0}^{B_0} \).

4) After each sequence in \( A_{n_1}^{B_1} \) has been assigned a codeword from \( C_{n_0}^{B_0} \), sequences not in \( A_{n_1}^{B_1} \) are assigned codewords in an arbitrary manner from the remaining codewords (which may or may not be in \( C_{n_0}^{B_0} \)).

As \( n \to \infty \), the probability of a “1” at the output of the L2 encoder is at most \((1 - \epsilon_1)p(n) + \epsilon_1n/n\), where:

1) the probability of a sequence being in \( A_{n_1}^{B_1} \) is at least \( 1 - \epsilon_1 \) (AEP);
2) as \( n \to \infty \), the number of 1’s in a codeword encoding a sequence from \( A_{n_1}^{B_1} \) is \( pn \) (strong law of large numbers);
3) the probability of a sequence not being in \( A_{n_1}^{B_1} \) is at most \( \epsilon_1 \) (AEP);
4) the number of 1’s in a codeword encoding a sequence not from \( A_{n_1}^{B_1} \) 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 \( p_n \) or \( H^{-1}(H) \), 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.

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 \( (X_1, X_2, \ldots, X_N) \), generated by the process. The symbols are coded, independently of any other symbols, to the \( \tilde{N} = NR \) bits represented by the binary random variables \( B_1, B_2, \ldots, B_n \). Clearly, as \( \tilde{N} \) (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/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., \( (X_1, X_2, \ldots, X_N) = f(B_1, B_2, \ldots, B_n) \). The entropy rate of the process \( \{B_i\} \) is given by

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

\[
\geq \lim_{n \to \infty} \frac{1}{n} H_f = \frac{1}{NR} H(X_1, X_2, \ldots, X_N) = \frac{H}{R}.
\]

\[
\Rightarrow \lim_{n \to \infty} \frac{1}{n} H(B_1, B_2, \ldots, B_n) \geq \frac{H}{R}. \quad \text{(C.1)}
\]
Define a function \( g \) on \((B_1, B_2, \ldots, B_n)\) as follows:

\[
(C_1, C_2, \ldots, C_n) = g(B_1, B_2, \ldots, B_n) \tag{C.2}
\]

where \( C_i = B_i \oplus B_{\text{prev}(i)} \). 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 \((C_1, C_2, \ldots, C_n)\) as \( B_i = C_i \oplus B_{\text{prev}(i)} \). Fig. 6 shows the relation between \( B_i, B_{\text{prev}(i)} \), and \( C_i \). The two delays in Fig. 6 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

\[
\lim_{n \to \infty} \frac{1}{n} H(C_1, C_2, \ldots, C_n) = \lim_{n \to \infty} \frac{1}{n} H(B_1, B_2, \ldots, B_n), \tag{C.3}
\]

From (C.1) and (C.3), we get

\[
\lim_{n \to \infty} \frac{1}{n} H(C_2, C_3, \ldots, C_n) \geq \frac{H}{R} \tag{C.4}
\]

Let \( p_c = \lim_{n \to \infty} (1/n) \sum_{i=1}^{n} C_i \), 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, \( p_c \) is the number of “1’s” in \((C_1, C_2, \ldots, C_n)\) for large \( n \). Substituting \( p_c \) for \( p_c \) and \( H/R \) for \( H \) in Lemma 2, we obtain

\[
H^{-1}(\frac{H}{R}) \leq p_c \leq 1 - H^{-1}(\frac{H}{R}). \tag{C.5}
\]

Since there are on the average \( R \) bits/symbol and \( C_i \) is ‘1’ if there was a transition at \( B_i \)

\[
T = p_c R. \tag{C.6}
\]

Multiplying (C.5) by \( R \) and employing (C.6), we have (2.5).

**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 \( 2^{kH} \) 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 \( H^{-1}(H/R) \) of being a 1. Thus, there are \( 2^{kH}H^{-1}(H/R) = 2^kH \) codewords (AEP). Hence, we have a codeword for each block of symbols. Since each bit in each codeword has probability \( H^{-1}(H/R) \) of being a 1, 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/symbol.

**REFERENCES**


