SIGNAL CODING FOR LOW POWER: FUNDAMENTAL LIMITS AND PRACTICAL REALIZATIONS

Sumant Ramprasad *Naresh R. Shanbhag* *Ibrahim N. Hajj*

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT

Transitions on high capacitance busses 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%.

1. INTRODUCTION

The on-chip dynamic power dissipation of CMOS circuits at a node is given by, $P_D = \frac{1}{2}TC_L V_{dd}^2 f$, 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 $[1,3,6,7]$ 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 lower and upper bounds on the expected transition activity for any coding algorithm. The fundamental bounds 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. We then present a communication-theoretic source-coding framework for the design of coding schemes to reduce transition activity. 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 at the bus. A framework to characterize low-power encoding schemes is developed based upon the source-channel coding view. In this 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 [5], whereby equivalence between computation and communication is being established.

2. BOUNDS ON TRANSITION ACTIVITY

In this section, we present achievable lower and upper bounds on the expected number of transitions. Theorem 1 bounds the number of transitions/symbol of a source with a certain entropy rate $\mathcal{H}$ given that each symbol is coded employing an expected number of R bits. The function $H(x)$ is defined on the real interval [0,1] as follows,

$$H(x) \quad = \quad -x\log_2 x - (1-x)\log_2(1-x). \qquad (2.1)$$

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

$$H^{-1}(x) = y, \text{ if } H(y) = x \text{ and } y \in [0, \frac{1}{2}]. \qquad (2.2)$$

Theorem 1 *Let,*

1. *$\mathcal{H}$ be the entropy rate of a process $\{X_i\}$,*
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, and*
4. *T be the expected number of transitions in the bits on the wires per symbol (i.e. $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} B_i \bigoplus B_{prev(i)}$ exists, where the function $prev(i)$ returns the index of the bit that immediately precedes B_i on the same wire and $\bigoplus$ is the exclusive-or operator)*

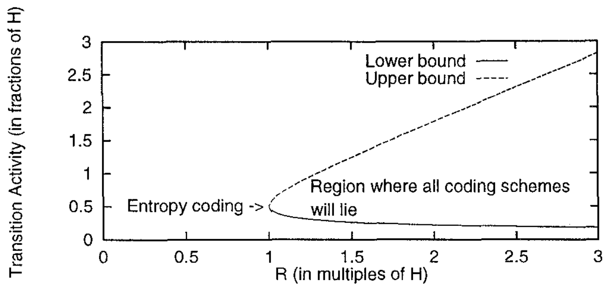

Figure 1. Lower and Upper bound on Transition Activity versus R

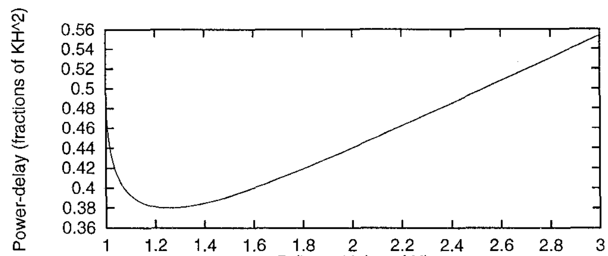

Figure 2. Lower bound on the power-delay product versus R for given $\mathcal{H}$

then,

$$H^{-1}(\frac{\mathcal{H}}{R})R \leq T \leq (1 - H^{-1}(\frac{\mathcal{H}}{R}))R \quad (2.3)$$

and the bounds in (2.3) are asymptotically achievable if $\{X_i\}$ 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 Figure 1. Any coding algorithm will need to reside in the region shown in Figure 1.

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 $R = 10\mathcal{H}$.

2.1. Applications of bounds on transition activity

In this section, we employ Theorem 1 to derive a lower bound on the power-delay product and also the lower-bound for the transition activity of any coding scheme encoding data from a given source.

2.1.1. 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. Hence the lower bound on the power-delay product, $PowerDelay_{min}$, given $\mathcal{H}$ and R, is given by,

$$PowerDelay_{min} = KH^{-1}(\frac{\mathcal{H}}{R})R^2, \quad (2.1)$$

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 2.

For given $\mathcal{H}$, we can find the R that minimizes $PowerDelay_{min}$ by equating the derivative of (2.1) with respect to R to 0. The value of R that minimizes $PowerDelay_{min}$ is found to be,

$$R_{min,power-delay} = 1.25392\ \mathcal{H}. \quad (2.2)$$

Thus, (2.2) indicates that a source 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.

2.1.2. 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,

$$\mathcal{H} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} - \frac{1}{16}\log_2\frac{1}{16} = \frac{15}{8}\text{ bits.} \quad (2.3)$$

Assume an average of $R = 3$ bits are employed to code a symbol. Thus $\frac{\mathcal{H}}{R} = \frac{5}{8}$, $H^{-1}(\frac{\mathcal{H}}{R}) = 0.156142$, and from Theorem 1, the bounds on transition activity are,

$$0.468426 \leq T \leq 2.531574. \quad (2.4)$$

3. PRACTICAL CODING SCHEMES

A generic communication system consists of a *source coder*, a *channel coder*, a *noisy channel*, a *channel decoder*, and a *source decoder*. The source coder (decoder) compresses (decompresses) the input data so that the number of bits required in the representation of the source is minimized. While the source coder removes redundancy, the channel coder adds just enough of it to combat errors that may arise due to the noise in the physical channel. In the present context, we consider the bus between two chips as the physical channel and the transmitter and receiver blocks to be a part of the pad circuitry, driving (in case of the transmitting chip) or detecting (in case of the receiving chip) the data signals. Furthermore, we will assume here that the signal levels are sufficiently high so that the channel can be considered as being noiseless. While this *noiseless channel* assumption is true for most systems today, this will not be the case for future systems where the signal swings will be lowered to reduce power. The noiseless channel assumption allows us to eliminate the channel coder.

The proposed communication-theoretic framework in Figure 3 is based upon the low-complexity source coder architecture. The function f_1 decorrelates the B bit input $x(n)$. Therefore, the prediction error, $e(n)$, is a function, f_1, of the current value of $x(n)$ and the prediction,

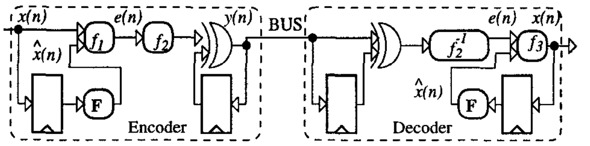

Figure 3. Low-Power Encoder-Decoder Framework

$\widehat{x}(n)$. The function f_1 could be a linear or a non-linear function of its arguments. The prediction, $\widehat{x}(n)$, is a function, $\mathbf{F}(x(n-1), x(n-2), \ldots, x(n-M+1))$, 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 code-word 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 3.

3.1. 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 *Exclusive-Or* 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 $\widehat{x}(n)$ 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 $e(n)$ 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 [6]. In the *pbm* function, the number of 1's in the input is reduced by assigning, as in [3], code-words with fewer 1's to the more frequently occuring 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 code-words 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*.

Table 1. Encoding Schemes

Encoding scheme	F	f_1	f_2
xor-pbm	*Identity*	*xor*	*pbm*
xor-vbm	*Identity*	*xor*	*vbm*
dbm-pbm	*Identity*	*dbm*	*pbm*
dbm-vbm	*Identity*	*dbm*	*vbm*
xor-inv (Bus-Invert)	*Identity*	*xor*	*inv*
dbm-inv	*Identity*	*dbm*	*inv*
inc-xor	*Increment*	*xor*	*Identity*

3.2. Encoding Schemes

The proposed encoding schemes are summarized in Table 1, where we have 7 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 occuring set of transitions in the original signal. The closest approach in literature to *xor-pbm* is [3] where signal samples having higher probability of occurrence are assigned code-words with fewer ON bits. In VLSI circuits, power dissipation depends on the number of transitions occuring at the capacitive nodes of the circuit. The *xor-pbm* scheme differs from [3] in two respects. It reduces the power dissipation by reducing the number of transitions by assigning fewer transitions to the more frequently occuring 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 Figure 3 can be employed to derive and improve existing coding schemes. For example, the Gray coding scheme can be derived by letting $f_1(x(n), \widehat{x}(n)) = x(n)$, f_2 be the Gray coding scheme, and removing the exclusive-or at the output of the encoder. The Bus-Invert [6] 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 [3] can be derived by letting $f_1(x(n), \widehat{x}(n)) = x(n)$, 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 Figure 4. Unlike the T0 scheme, the

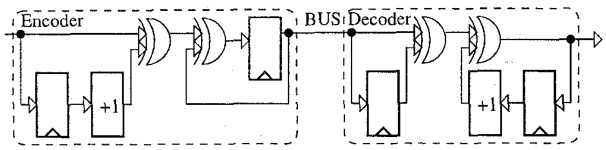

Figure 4. Encoder and Decoder for *inc-xor*

Table 2. Description of data sets

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

inc-xor scheme does not require an extra bit, has a shorter critical path, and provides the same or more reduction in transitions.

The performance of the above coding schemes with the data sets in Table 2 is shown in Table 3.

3.3. Comparison of Encoding Schemes

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

Assume that uniformly distributed, independent data is transmitted over a B-bit bus. Hence $\mathcal{H} = B$. In schemes such as Bus-Invert coding [6], 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 1 bit of redundancy. Thus $R = \mathcal{H} + 1 = B + 1$ and hence, from Theorem 1, the expected transition activity is bounded by,

$$(B+1)H^{-1}(\frac{B}{B+1}) \leq T \leq (B+1)(1 - H^{-1}(\frac{B}{B+1})). \quad (3.1)$$

If $B = 8$ then Bus-Invert achieves a transition activity of

Table 3. Percentage Reduction in Transition Activity

Data	*xor-pbm*	*xor-vbm*	*dbm-pbm*	*dbm-vbm*	*xor-pbm* with *pbm* opt. for	Redn.	Adapt. scheme
A3	32	24	**36**	29	A3	32	24
A4	38	29	**41**	30	A3	37	29
A7	41	32	**45**	25	A3	39	33
CO	26	4	**28**	**28**	A3	20	4
V1	39	34	**44**	44	V2	38	39
V2	33	26	**39**	39	V2	33	33
V3	33	25	**40**	39	V2	33	33
R1	**1**	0	1	0	V2	0	0
PS	**43**	22	38	25	PS	**43**	42

3.269 transitions/8 bits. The lower bound from Theorem 1 is 2.4506 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 $\frac{B}{B+1}$ approaches 1 and T approaches $\frac{B}{2}$. 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.

3.3.1. *Probability Based Coding*

We use the i.i.d. source in sub-section 2.1.2 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 per symbol is,

$$\begin{aligned} T &= 0 * \frac{1}{2} + 1 * \frac{1}{4} + 1 * \frac{1}{8} + 1 * \frac{1}{16} + 2 * \frac{1}{16} \\ &= 0.5625 \text{ transitions/symbol}, \end{aligned}$$

which is within 17% of the lower bound in (2.4). We can further reduce the number of transitions by applying probability based coding to a block of two symbols. The expected number of transitions per 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 8.

ACKNOWLEDGMENT

This work was supported by DARPA contract DABT63-97-C-0025, NSF CAREER award MIP-9623737, and Joint Services Electronics Program under contract #00014-96-1-0129

REFERENCES

[1] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, "Asymptotic Zero-Transition Activity Encoding for Address Busses in Low-Power Microprocessor-Based Systems," *Great Lakes VLSI Symposium*, pp. 77–82, Urbana IL, March 13–15, 1997.

[2] T. M. Cover and J. A. Thomas, "Elements of Information Theory," John Wiley & Sons, 1991.

[3] D. W. Faulkner, "PCM Signal Coding," *U.S. Patent no. 5,062,152*, October 1991.

[4] D. Marculescu, R. Marculescu, and M. Pedram, "Information theoretic measures for power analysis," *IEEE Trans. CAD*, vol. 15, no. 6, pp. 599–610, June 1996.

[5] N. R. Shanbhag, "A mathematical basis for power-reduction in digital VLSI systems," *IEEE Trans. CAS II*, vol. 44, no. 11, pp. 935–951, November 1997.

[6] M. R. Stan and W. P. Burleson, "Bus-Invert Coding for Low-Power I/O," *IEEE Trans. VLSI*, vol. 3, no. 1, pp. 49–58, March 1995.

[7] M. R. Stan and W. P. Burleson, "Limited-Weight Codes for Low-Power I/O," *Int. Workshop on Low Power Design*, pp. 209–214, Napa CA, April 1994.