Low-Power Bus Encoding Using An Adaptive Hybrid Algorithm

Avnish R. Brahmbhatt, Jingyi Zhang, Qing Wu, Qinru Qiu
Department of Electrical and Computer Engineering
Binghamton University, State University of New York
Binghamton, New York 13902
{abrahmb1, jzhang5, qwu, qqiu}@binghamton.edu

ABSTRACT

In this paper, we propose an adaptive low-power bus encoding algorithm based on weighted code mapping (WCM) and the delayed bus technique. The WCM algorithm transforms an original bus data vector to a low-energy code through one-to-one mapping. The code mapping is determined by the data probabilistic distribution in the original sequence. The WCM algorithm considers both the self and coupling capacitance of the bus wires. In addition, we found that applying the delayed-bus technique can further reduce the bus energy. A window-based adaptive encoding algorithm is proposed to improve the energy saving by adaptively changing the code mapping when significant data changes are detected. Experimental results show that the proposed algorithm outperforms the existing heuristic bus encoding algorithms by 20~60% in energy dissipation.

Categories and Subject Descriptors

B.7.1 [**Hardware**]: Integrated Circuits – *Types and Design Styles*

General Terms

Algorithms, Performance, Design

Keywords

low power, bus encoding, adaptive algorithm, data probabilistic distribution, weighted code mapping, delayed bus, window based change detection

1. INTRODUCTION

As technology scales down to deep submicron, distance between adjacent wires is decreasing, leading to the increase in inter-wire capacitances. Significant amount of power is dissipated in charging and discharging of parasitic capacitance between adjacent lines. The crosstalk energy is becoming the major component in bus energy[1]-[3].

Extensive research works have been done for the optimization of bus energy dissipated on the coupling capacitances. One popular technique is to encode data before transmission and decode the code words at the receiver's side. Most of the bus encoding algorithms needs extra bit lines to generate a larger code word space, from which codes with smaller energy dissipation can be selected.

Existing bus encoding methods can be divided into two categories: heuristic and probabilistic-based. Among the heuristic algorithms, authors of [2] presented a "forbidden pattern" coding scheme, which reduces crosstalk by eliminating certain high-energy patterns in the transmitted data. Reference [4] introduces a "forbidden transition" coding scheme for crosstalk reduction, in which no adjacent bits can switch in opposite directions. An even-odd bus-inverting algorithm was proposed in [3], which divides the data vector into groups of even bits and odd bits. Both groups could be inverted independently to reduce the cross-coupling energy. Authors of [11] introduced the "delayed bus" algorithm that spaces out the timing of the transitions on adjacent bus wires to reduce bus energy. The advantage of the heuristic algorithms is that they are simple for implementation, with low hardware and computation overheads. The shortcoming is that they do not consider the probabilistic distribution of the transmitting data, leading to limited energy savings.

In the category of probabilistic-based bus encoding methods, the transition pattern coding scheme (TPC) [6], works effectively on general bus model with both the coupling capacitance and the line-to-ground capacitance. The approximately optimal coding (AOC) scheme is introduced as a reduced version of TPC to reduce the runtime of searching. The TPC algorithm provides the optimal bus encoding, given the probabilistic distribution of the data. The shortcomings are that its effectiveness relies on the accurate information about the data sequence, and the hardware and computation overheads are high and increase exponentially with the increase of bus width [6][7].

In this paper, we propose a new adaptive low-power bus-encoding scheme. The proposed weighted code mapping (WCM) algorithm generates a one-to-one mapping from the original data to the coded data. In this algorithm, we first generate a set of codes with low bus-energy dissipation. Then based on the probabilities of the original data, we derive the mapping for minimum bus energy. In addition, we propose a hybrid algorithm that combines WCM algorithm with the "delayed bus" approach to further reduce energy dissipation. We also propose an adaptive algorithm that dynamically adjusts the encoding based on the change of the data probabilistic distribution. For proposed algorithm, the complexity of code searching and the hardware/computation overheads are significantly reduced, comparing to TPC or AOC. Comparing to the existing heuristic algorithms, the proposed algorithm provides more energy reduction by both eliminating more crosstalk transitions and considering the probabilistic of the input data.

The rest of the paper is organized as follows. Section II introduces the background on the DSM bus model. The proposed algorithms are presented in section III. In section IV, experimental results are presented and analyzed in detail. Section 5 gives the summaries.

2. BACKGROUND

In general, a bus may consist of one set of parallel wires or more with repeaters between them. The bus wires are usually laid in parallel and the capacitive coupling between nonadjacent lines is weak compared to that between adjacent lines. The model of the capacitive parasitic of a bus can be simplified to the circuit shown

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC 2006, July 24–28, 2006, San Francisco, California, USA.
Copyright 2006 ACM 1-59593-381-6/06/0007…$5.00.

in Figure 1, where C_I is the inter-line capacitance, while C_L is the line-to-ground capacitance.

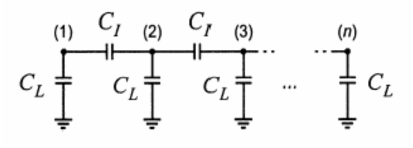

Figure 1 The DSM bus capacitive model.

The conductance matrix of the network can be written as ([6][8])

$$\overline{C} = \begin{bmatrix} 1+\lambda & -\lambda & 0 & \dots & 0 & 0 \\ -\lambda & 1+2\lambda & -\lambda & \cdots & 0 & 0 \\ 0 & -\lambda & 1+2\lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdot & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1+2\lambda & -\lambda \\ 0 & 0 & 0 & \cdots & -\lambda & 1+2\lambda \end{bmatrix} \cdot C_L$$

where λ is called the *capacitance factor,* which is calculated as $\lambda = C_I / C_L$. Let $E(V^{old}, V^{new})$ denote the total energy drawn from the power supply during the transition from data vector V^{old} to V^{new}. $E(V^{old}, V^{new})$ can be as the following

$$E(V^{old}, V^{new}) = (V^{new})^T \overline{C} (V^{new} - V^{old})$$

Let w_1, w_2, …, w_{2^n} be the set of all possible binary vectors transmitted on a n-bit bus. The *energy cost matrix* can be defined as

$$\mathbf{E} = \left[E(w_i, w_j) \right]_{i,j=1}^{M} .$$

3. ADAPTIVE ALGORITHM WEIGHTED CODE MAPPING

In this section, we will first introduce the bus encoding algorithm based on weighted code mapping, followed by the hybrid algorithm and the window-based adaptive encoding method.

A. Bus encoding using weighted code mapping

One of the characteristics of energy cost matrix is that some columns consistently have lower total summation than others. It indicates that transmitting certain codeword always dissipate less energy than transmitting others, no matter what is the previous vector on the bus. Let $Eng(i)$ denote the summation of all the entries in the ith column plus the summation of all the entries in the ith row of the energy cost matrix. It can be proved that the value of $Eng(i)$ is proportional to the expected bus energy per clock if vector v_i is transmitted [12].

Based on the above observation, we designed coding algorithm called the Weighted Code Mapping (WCM) algorithm. Let m and n denote the width of the binary vectors before and after encoding and $n = m + a$, where a is the number of redundant wires. We assume that the probabilistic distribution of the input data vector is known. The WCM algorithm finds a set of 2^m codewords from the set of 2^n binary vectors such that the expected steady-state bus energy is minimized. A one-to-one mapping is then formed from the input data vectors to these 2^m codes based on the probability information.

Given an n-bit binary vector $a_0a_1a_2...a_n$, an *intra-vector-switch* (*IVS*) is defined as two adjacent bits a_i and a_{i+1} with $a_i \neq a_{i+1}$. Given a codeword w, its *intra-vector-switch number*, which is denoted as $IVS(w)$, is the number of occurrence of the pattern "01" and "10" in this codeword. For example, $IVS(0000100) = 2$. It can be proved that the $Eng(i)$ is solely determined by $IVS(w_i)$.

Theorem 1: $Eng(i) > Eng(j)$ iff $IVS(w_i) > IVS(w_j)$.

Based on theorem 1, the problem of searching for a set of codewords with the minimal $Eng(i)$ values can be transformed into the problem of generating a set of binary vectors with the minimal number of intra-vector-switches. It is easy to see that there are $2^* C_{n-1}^{ivs}$ codeword that has *ivs* intra-vector-switches. Figure 2 gives the pseudo code of the improved WCM algorithm. The following examples will show the working of our algorithm.

```
Weighted Code Mapping:-
1. Set codeword array A[2^m]=0;
2. ivs=0;
3. C=0;
4. id = 0;
5. While (|A|)<2^m {
6.   If (C== 2* $C_{n-1}^{ivs}$ ) {
7.      ivs++;
8.      C=0;}
9.   Generate the next binary vector w with IVS(w)=ivs;
10.  A[id++] = w;
11. }
12. Sort the input data based on their probability;
13. Map the highest probability data to the code with the smallest IVS;
```

Figure 2 The WCM algorithm based on pattern generation.

Example 1 Consider $m = 2$, $a = 1$, $\lambda = 3$, probability of data is p {00, 01, 10, 11} = {0.1, 0.4, 0.4, 0.1} and the data stream to be transmitted is {01, 10, 01, 10, 00, 11, 10, 01, 01, 10}. From WCM algorithm, codewords are selected as {000, 111, 001, 011}, and the mapping is 00-001, 01-111, 10-000, 11-011. Energy matrix for $m = 2$, $a = 0$ and $\lambda = 3$ is shown below.

$$E = \begin{bmatrix} 0 & 4 & 4 & 2 \\ 0 & 0 & 7 & 1 \\ 0 & 7 & 0 & 1 \\ 0 & 3 & 3 & 0 \end{bmatrix} \cdot C_L$$

We can calculate total energy without coding as:

$$01 \underbrace{\to}_{7} 10 \underbrace{\to}_{7} 01 \underbrace{\to}_{7} 10 \underbrace{\to}_{0} 00 \underbrace{\to}_{2} 11 \underbrace{\to}_{3} 10 \underbrace{\to}_{7} 01 \underbrace{\to}_{0} 01 \underbrace{\to}_{7} 10 \quad = 40^*C_L$$

We can calculate total energy with WCM encoding as:

$$111 \underbrace{\to}_{0} 000 \underbrace{\to}_{3} 111 \underbrace{\to}_{0} 000 \underbrace{\to}_{4} 001 \underbrace{\to}_{4} 011 \underbrace{\to}_{0} 000 \underbrace{\to}_{3} 111 \underbrace{\to}_{0} 111 \underbrace{\to}_{0} 000$$

$= 17^*C_L$

Total energy saving is (40-17)/40*100 = 57.5%.

Example 2 For this example we will take the same inputs as used for example 1, to find codewords will start from $ivs = 0$, so there are two possible codewords 000 and 111, still we need two more codes so ivs will be incremented by 1 and for $ivs = 1$, the $2^* C_{n-1}^{ivs} = 4$ and codewords are {001, 011, 100, 110}. We select the first two of them and form the complete set of codewords as {000, 111, 001, 011}, which is same as what we obtained for WCM algorithm.

For more information about WCM algorithm please refer to reference [12].

If the input data follows uniform distribution, after code mapping, the expected bus energy per clock can be calculated as $E_{\exp} = \frac{1}{2^m} \sum_{i,j} E(w_i, w_j)$, where w_i and w_j belongs to the set of selected codewords. We need to point out here that the set of codewords that have the minimum *Eng* may not be the set of codewords that gives the minimum E_{exp}. However, these two are very close. The second problem is equivalent to search for a minimum weight clique, which is NP hard. Experiments are carried out to search for the truly optimal codewords that minimizes E_{exp}. Parallel genetic algorithm (GA) using island multi-deme model [10] is used to search for the optimal codewords. Each individual in the population corresponds to a valid selection of codewords. In our search algorithm, the population size is set to 8000. The mutation probability is set to 4% and at each generation 10% of the best individuals are kept. The maximum generation is set to 100000. The program runs parallel on 8 cluster computers. Comparing the GA to WCM, the improvement that we have observed is about 6% for 4-bit bus and under 1% for 5+ bus wires.

B. Hybrid of WCM and delayed bus algorithm

The delayed bus algorithm proposed in [11] works on the principle of eliminating opposite ("2λ") transitions on adjacent bus wires. The algorithm delays all the 0-to-1 transitions on a bus by certain amount of time to avoid the "2λ" transitions. For example, if a bus is making a transition from 1010 to 0101, the second and fourth bus line will be delayed. Therefore the resulting data transitions on the bus are 1010-0000-0101, with no more cross-coupling energy.

Even though the WCM algorithm works better than other heuristic algorithms (which will be shown in Section 4), as the bus width increases there are possibilities to have "2λ" transitions that will lead to more energy dissipation. For example, consider a 16-bit bus that can have transition from 1010100000000000 to 0101010000000000 and hence "2λ" transitions. With large bus width, this type of codes can be easily selected as *ivs = 5 and 6*, respectively. The chances of having "2λ" transitions are even higher if the input data follows uniform distribution. In order to further save energy in such cases, we designed a hybrid bus encoding algorithm that applies WCM first and followed by the delayed bus technique.

C. Window-based adaptive bus encoding

One limitation of the WCM algorithm is that its optimality relies on the information of the probability distribution of the data stream. In real applications, data usually arrives in the form of continuous unbounded streams. It is therefore, unrealistic to allow our coding algorithms enough memory capacity to store the full history of the data stream. Furthermore, the data distribution may differ dramatically over time. It is desirable that we detect the change of distribution in the data stream and use different code mapping for segments in the data stream with different probabilistic distributions.

In this work, we propose a window-based adaptive encoding algorithm for best energy saving on a bus. The pseudo code of the algorithm is given in Figure 3.

```
Window-based adaptive bus encoding
1: WinSize = the size of detection window;
2: WinX = the first WinSize data in the stream;
3: Perform code mapping based on the data statistics of WinX;
4: while ( not the end of stream) {
5:          WinY =  the next WinSize data in the stream;
6:          if (Distance (WinX, WinY) < Threshold)
7:                    Do nothing;
8:          else {
9:                    Perform code mapping based on the data
10:                   statistics of WinY;
11:                   WinX = WinY;
12:             }
13: }
```

Figure 3 An adaptive bus encoding algorithm.

In our algorithm, the size of the window will affect the accuracy and speed of detecting data distribution changes. Using small windows can detect sudden, distinct changes while using large windows can detect subtle changes over long periods of time. When selecting the size of the window, our goal is to make it just large enough to reflect the probabilistic changes accurately, while we still can detect a change fast enough when it happens.

The data stream is divided into segments (windows) of size *WinSize*. *WinX* is used as a reference window. It is always the most recent window in which a change in the data stream has been detected, with respect to the previous reference window. *WinY* is the moving window that moves from one segment to the next. Every *WinY* changes, the "Distance" function is used to evaluate the difference in probabilistic distribution between the data in *WinY* and *WinX*. If the result is within the pre-defined "Threshold", we will keep the current bus encoding scheme. If the distance is larger than the threshold, we will re-calculate the mapping between data and code based on the data probabilities in WinY. At the same time, we assign *WinY* to *WinX* as the new reference window.

In this algorithm, the choices of the distance function and the threshold are crucial to the out come of the encoding. Our goal is to minimizing the times that we need to re-calculate the encoding, while achieve the best possible bus energy reduction. We have tried several well-known probabilistic test methods for our application and eventually determined to use the method proposed in [9]. The definition of the distance function and the procedure in determining the threshold is introduced briefly in the following paragraphs.

Let P_1, P_2 be two probabilistic distributions over a measure space, and define α-distance between P_1 and P_2 is defined as:

$$d_\alpha(P_1, P_2) = 2 \sup_{A \in \alpha} |P_1(A) - P_2(A)|$$

where α is a collection of measurable sets.

The α-distance represents the largest change in probability of a set from the measure space, indicating the significance of the change. In our algorithm, we use the relative discrepancy, which is a variation of the α-distance as our distance function. It can be written as:

$$\sup_{A \in \alpha} \frac{|P_1(A) - P_2(A)|}{\sqrt{\min\left\{\frac{P_1(A)+P_2(A)}{2}, \left(1 - \frac{P_1(A)+P_2(A)}{2}\right)\right\}}}$$

where $P_1(A)$ and $P_2(A)$ are the data distribution functions of WinY and WinX respectively.

Next, we need to determine the value of the threshold. Let F be the maximum value of the distances over all the window pairs of WinY and its reference window WinX before any detection of data change. The threshold is set to be the 1-p quintile of the distribution function of F. If the data stream is generated independently by any fixed continuous probability distribution G, then the distribution of F does not depend on G [9]. It means that in theory, the threshold can be calculated from n samples of data from any probabilistic distributions, such as the uniform distribution. However, real-time multimedia data steams are rarely follow any distribution, which requires non-parametric test on the data under no assumptions on the form of the distributions. Then the threshold can be obtained more accurately by running the simulations over several training sequences of randomly sampled real multimedia data.

4. EXPERIMENTAL RESULTS

In our experimental setups, we first compare the WCM and hybrid algorithms with other existing heuristic algorithms. Random data sequences of 4, 8 and 16-bit wide are generated following three different types of random distributions: triangular distribution (sequences T-4, T-8 and T-16), uniform distribution (sequences U-4, U-8 and U-16 and normal distribution (sequences N-4, N-8 and N-16). Bus energy comparisons are done for the bus-inverting (BI) [5], the odd-even-inverting (OEI), the forbidden pattern (FP), the original delayed bus algorithm (DB), the WCM algorithms with one redundant bit (WCM1) and two redundant bits (WCM2), and the correspondent hybrid algorithms (HYB1, HYB2). The simulation results for $\lambda=5$ are shown in Table 1. In the table, the last column shows the percentage improvement of the HYB2 over the best of BI, OEI, FP and DB. Table 2 shows the results for $\lambda=10$.

Table 1 Comparisons of bus encoding algorithms (λ=5).

Data	Bus energy per vector ($V_{dd}^2*C_L$)								Impv.
	BI	OEI	FP	DB	WCM1	WCM2	HYB1	HYB2	%
T-4	7.34	8.31	4.25	6.75	4.23	3.18	4.05	3.12	26.5
U-4	7.38	8.49	5.37	6.90	5.43	4.04	5.08	3.77	29.8
N-4	7.52	8.62	7.50	7.64	4.24	3.24	4.08	3.17	57.8
T-8	16.21	17.06	14.86	15.61	12.58	10.72	11.70	10.23	31.2
U-8	16.32	17.31	15.95	15.73	14.52	12.01	13.13	11.31	28.1
N-8	16.69	17.39	15.53	16.65	11.32	9.71	10.71	9.35	39.8
T-16	35.59	36.39	38.14	33.66	30.87	27.57	27.84	25.49	24.3
U-16	35.68	36.57	34.51	33.80	33.48	29.59	29.55	26.99	20.2
N-16	34.39	36.10	35.67	34.14	24.90	22.72	23.44	21.66	36.5

Table 2Comparisons of bus encoding algorithms (λ=10).

Data	Bus energy per vector ($V_{dd}^2*C_L$)								Impv.
	BI	OEI	FP	DB	WCM1	WCM2	HYB1	HYB2	%
T-4	13.89	15.67	8.00	12.60	7.48	5.41	7.12	5.29	33.9
U-4	13.92	15.99	9.12	12.80	9.90	7.13	9.20	6.58	27.9
N-4	14.24	16.25	14.28	14.30	7.52	5.54	7.19	5.40	62.1
T-8	31.25	32.88	26.82	29.62	23.15	19.45	21.32	18.41	31.3
U-8	31.39	33.31	29.18	29.77	27.15	21.97	24.29	20.47	29.9
N-8	32.32	33.52	28.71	31.76	20.64	17.44	19.35	16.70	41.9
T-16	68.39	69.93	70.86	63.69	57.83	51.17	51.69	46.97	26.3
U-16	68.50	70.25	63.19	63.86	63.16	55.24	55.15	49.98	20.9
N-16	66.08	69.42	66.17	64.57	45.82	41.53	42.92	39.42	38.9

Next, we compare the bus energy consumptions of the "adaptive hybrid" (Adapt_HYB) algorithm with the "global hybrid" (Global_HYB) algorithm. In this setup, each benchmark sequence contains four subsequences with different data characteristics. For example, the "U-8-4" is an 8-bit data sequence that consists of four subsequences of uniformly distributed data with different means. The "Image-8-4" is an 8-bit sequence with subsequences selected from four different images. The "Mix-8-4" is a sequence with one image sequence, two audio sequences and one video sequence. The Other sequences are generated in similar ways. Note that the "Global_HYB" algorithm calculates the encoding just once, based on the probabilistic distribution of the whole sequence, while the "Adapt_HYB" algorithm may calculate the encoding multiple times, based on the variations of the data probabilistics. Table 3 shows the experimental results for a=1 and λ=5. The results show average saving of about 23% in bus energy for the "adaptive hybrid" algorithm.

Table 3 Comparison of adaptive and global coding (λ=5).

Data	Energy per vector ($V_{dd}^2*C_L$)		# of code re-calculation	Improvement (%)
	Global_HYB	Adapt_HYB		
T-8-4	12.46	8.62	3	30.8
U-8-4	9.61	7.13	3	25.8
N-8-4	10.86	7.49	3	31.0
Image-8-4	7.80	7.47	8	6.2
Video-8-4	8.78	7.89	3	10.1
Audio-8-4	2.61	1.57	4	40.0
Mix-8-4	9.13	7.88	3	13.7

5. CONCLUSIONS

We have proposed an adaptive hybrid bus encoding algorithm that generates one-to-one mapping from the original data to the coded data, based on the probabilistic distribution information. The algorithm also adopts the delayed bus technique to further reduce the bus energy consumption. A window-base change detection algorithm is proposed to dynamically re-calculate the bus coding based on the input distribution. Experimental results show good energy savings by the proposed bus encode method.

REFERENCES

[1] S. R. Sridhara, A. Ahmed, and N. R. Shanbhag, "Area and Energy-Efficient Crosstalk Avoidance Codes for On-Chip Buses,", *Proc. Of IEEE International Conference on Computer Design*, 2004.

[2] C. Duan and A. Tirumala, "Analysis and Avoidance of Cross-talk in On-Chip Buses," *Hot Interconnects 9*, pp. 133-138, Aug. 2001.

[3] N. K. Samala, D. Radhakrishnan and B. Izadi, "A Novel Deep Sub-micron Bus Coding for Low Energy," *Proc. of the International Conference on Embedded Systems and Applications*, June 2004.

[4] B. Victor and K. Keutzer, "Bus Encoding to Prevent Crosstalk Delay," *IEEE/ACM International Conference on Computer Aided Design*, 2001.

[5] M.R. Stan and W.P. Burleson, "Bus-Invert Coding for Low Power I/O," IEEE Trans. VLSI Syst., pp. 49–58, Mar 1995.

[6] Sotiriadis, P., A. P. Chandrakasan, "Bus Energy Reduction by Transition Pattern Coding Using a Detailed Deep Submicrometer Bus Model," *IEEE Transactions on Circuits and Systems*, pp. 1280-1295, October 2003.

[7] L. Xie, P. Qiu, and Q. Qiu, "Partitioned Bus Coding for Energy Reduction, " *Proc. of Asia and South Pacific Design Automation Conference*, Jan. 2005.

[8] Sotiriadis P., A. P. Chandrakasan, " A Bus energy model for deep sub-micron technology," IEEE Trans. VLSI Syst., vol.. 10, pp. 341-350, June 2002.

[9] Daniel Kifer, Shai Ben-David, Johannes Gehrke, "Detecting Change in Data Streams", *Proceedings of the 30th VLDB Conference*, 2004.

[10] E. Cantu-Paz, "A Survey of Parallel Genetic Algorithms," Calculateurs Paralleles, Reseaux et Systems Repartis, Vol. 10, No. 2.

[11] Maged Ghoneima and Yehea Ismail, "Delayed Line Bus Scheme: A Low-Power Bus Scheme for Coupled On-Chip Buses," Low Power Electronics and Design, 2004. ISLPED'04. pp. 66-69.

[12] A.R. Brahmbhatt, J. zhang, Qinru Qiu, and Q. Wu, "Adaptive Low-Power Bus Encoding Based on Weighted Code Mapping," *Proc. of IEEE International Symposium on Circuits and Systems*, May 2006.