Partial Bus-Invert Coding for Power Optimization of System Level Bus

Youngsoo Shin, Soo-Ik Chae, and Kiyoung Choi
School of Electrical Engineering
Seoul National University
Seoul 151-742, Korea

Abstract

We present a partial bus-invert coding scheme for power optimization of system level bus. In the proposed scheme, we select a sub-group of bus lines involved in bus encoding to avoid unnecessary inversion of bus lines not in the sub-group thereby reducing the total number of bus transitions. We propose a heuristic algorithm that selects the sub-group of bus lines for bus encoding. Experiments on benchmark examples indicate that the partial bus-invert coding reduces the total bus transitions by 62.6% on the average, compared to that of the unencoded patterns.

1 Introduction

Recently, power consumption has been a critical design constraint in the development of digital systems due to widely used portable systems such as cellular phones and PDAs, which require low power consumption with high speed and complex functionality. Although the power consumption of a system can be reduced at various phases of the design process from system level down to process level, optimization at higher level can provide more power saving. Among the architectural components at the system level, buses that interconnect subsystems are important components, which consume a lot of power. Especially, power consumption for off-chip driving can reach up to 70% of the total chip power, where the bus transition is the most dominant factor [1]. Therefore, a considerable amount of power can be saved by reducing the power consumption of bus.

In this paper, we propose a new bus encoding scheme, called *Partial Bus-Invert (PBI)* coding, where the conventional bus-invert (BI) coding [2] technique is used but the technique is applied only to a selected subset of bus lines. We select such subset statically assuming that the information about the sequence of memory access patterns is available after the algorithm of an application is specified. Consequently, our focus is on special-purpose applications such as signal and image processing, which are commonly implemented as ASICs and off-chip memories connected by buses. We propose a heuristic algorithm that exploits both transition correlation and transition probability in order to find a subset of bus lines such that the total number of bus transitions are minimized when only the subset is encoded by the BI coding. Experimental results show that for benchmark examples the PBI coding for selected bus lines reduces the number of total bus transitions by 62.6% on the average, compared to that of the unencoded patterns.

ISLPED 98, August 10-12, 1998, Monterey, CA USA

2 Related Work and Motivation

The bus-invert code [2] is well suited to a data bus in a general-purpose system such as the one employing a microprocessor. Exploiting the fact that instruction addresses generated by a processor are often sequential, the T0 code [3] reduces the transitions by freezing the address lines when consecutive patterns are found to be sequential. In special-purpose applications where the address patterns are less sequential, the characteristics of address patterns can be exploited to efficiently reduce bus transitions. The Beach Solution [4] makes clusters of bus lines based on statistical information of address patterns and then generates an encoding function for each cluster such that the encoded version of each cluster results in less transitions.

The behavior of data addresses is somewhat different from that of instruction addresses or data. They are less sequential than instruction addresses. In case of some memory-intensive applications such as image processing algorithms, it is nearly out of sequence. However we can hardly assume that data addresses are random even though they are more random than instruction addresses. Usually, the signal probability and/or transition probability of some of bus lines are biased toward 0 or 1, that is, some of bus lines are far from random.

The motivation of the PBI coding is based on the observation that all the previously proposed coding schemes take the entire bus lines into account for bus encoding. However, the overhead of the encoding/decoding circuit increases with the number of bus lines involved in bus encoding. In the PBI coding, we attain two goals at the same time: minimizing the number of bus lines involved in bus encoding thereby minimizing the overhead and minimizing the total number of bus transitions.

3 Partial Bus-Invert Coding

3.1 Problem Formulation

In the BI coding, if the Hamming distance between the present pattern and the last pattern of the bus is larger than $n/2$, where n is the bus width, the present pattern is transmitted with each bit inverted. A redundant bus line, called *invert* line, is required to signal the receiver side whether the bus is inverted or not.

Now, let's consider encoding m out of n bus lines leaving the remaining bus lines unencoded. For the patterns randomly distributed in time and mutually independent in space, the more bus lines are involved in the BI coding, the more reduction in bus transitions can be obtained. Specifically, let $E(m)$ be the expected number of transitions per encoded pattern when we take m out of n bus lines for the BI coding while leaving the remaining bus lines unencoded. It can be shown that $E(m)$ is given by

$$E(m) = \frac{n}{2} - \sum_{i=m/2+1}^{m} (2i - m - 1) C_m^i \left(\frac{1}{2}\right)^m. \qquad (1)$$

The function monotonically (but not strictly) decreases with m but the amount of decrease is saturated as m increases. Therefore, we can obtain the maximum transition reduction when all the bus lines are involved in the BI coding, which is the case of the conventional BI coding. However, the monotonicity of the function does not hold when the behavior of patterns deviate from random distribution and mutual independence. In other words, the minimum of the function exists in between $m = 0$ and $m = n$.

Concerning data address buses, there is another reason why we should not adopt the conventional BI coding. It is the fact that usually some of the bus lines are *far* from random and it is inefficient to take those lines into account for the BI coding. However, it is difficult to determine quantitatively the criterion of how far is enough not to include them in the BI coding. The decision problem together with the non-monotonicity of the difference forms the following optimization problem: Given data address patterns of special-purpose applications, select a sub-group of bus lines for the BI coding such that the total power consumption in the bus (or including that of encoding/decoding circuit) is minimized.

3.2 Overview

In the PBI coding, we partition a bus $\mathcal{B}$ into two sub-buses based on the behavior of patterns transferred. More precisely, we are given a bus $\mathcal{B}$=$(b^0, b^1, \ldots, b^{n-1})$, which transfers a sequence of patterns $B_i = (b_i^0, b_i^1, \ldots, b_i^{n-1})$, where i is the time index, n is the bus width, and b_i^j is the value of a bus line b^j at time i. We partition $\mathcal{B}$ into a selected sub-bus $\mathcal{S}$ and the remaining sub-bus $\mathcal{R}$ such that $\mathcal{S}$ contains bus lines having higher transition correlation and/or higher transition probability and $\mathcal{R}$ contains the remaining bus lines. Because the bus lines in $\mathcal{R}$ have low correlation with those in $\mathcal{S}$ and low transition activity, they don't need to be involved in the BI coding. Inverting the lines in $\mathcal{R}$ will rather increase the transition activity than decrease it. Therefore, by applying the BI coding only to sub-bus $\mathcal{S}$, we can reduce the hardware for the BI coding as well as increase the gain of the BI coding.

Once $\mathcal{B}$ is partitioned, the PBI coding is performed as follows: We compute the Hamming distance between S_i' and S_{i+1}, where S_i' is an encoded version of S_i, including the *invert* line; if it is larger than $|\mathcal{S}|/2$, set the *invert* line to 1 and invert the lines in S_{i+1} without inverting the lines in R_{i+1}. Otherwise, set *invert=0* and let B_{i+1} uninverted.

3.3 Selection Algorithm of the Sub-bus

The performance of the PBI coding heavily depends on the selection of the sub-bus $\mathcal{S}$ for the BI coding. Unfortunately, it is intractable to find an optimum set $\mathcal{S}_{opt} \subseteq \mathcal{B}$ such that the PBI coding for $\mathcal{S}_{opt}$ results in the minimum number of total transitions. We propose a heuristic algorithm that explores only n configurations to find the one which results in minimal number of total transitions. To this end, we exploit both transition correlation and transition probability. For j-th bus line, the *transition encoding* is defined as

$$t_i^j = \begin{cases} 1 & \text{if } b_{i-1}^j \neq b_i^j \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The transition correlation coefficient for two bus lines (j-th and k-th) is defined by

$$\rho_{jk} = \frac{K_{jk}}{\sigma_j \sigma_k}, \quad (3)$$

where σ_j is the standard deviation of t^j and K_{jk} is the covariance of t^j and t^k.

Algorithm *Select_bus_lines*
begin
L1: Compute the transition probability of each line, tp_j;
L2: Compute ρ_{jk} of each pair of line;
L3: $\mathcal{S} = \{\}$, $\mathcal{R}$=$\{b^0, b^1, \ldots, b^{n-1}\}$;
L4: Initialize the configuration set $C = \{\}$;
L5: Select b^i with the highest transition probability;
L6: $\mathcal{S}$=$\{b^i\}$, $\mathcal{R}$=$\mathcal{R}-\{b^i\}$, $C = C \cup \{(\mathcal{S}, \mathcal{R})\}$;
L7: **while** $\mathcal{R} \neq \{\}$ **do**
L8: Select b^j such that $\frac{\sum_{b^k \in \mathcal{S}} \rho_{jk}}{|\mathcal{S}|} + tp_j$ is a maximum;
L9: $\mathcal{S} = \mathcal{S} \cup \{b^j\}$, $\mathcal{R} = \mathcal{R} - \{b^j\}$, $C = C \cup \{(\mathcal{S}, \mathcal{R})\}$;
end do
L10: Count the number of total transitions after PBI coding for each configuration in C;
L11: Select the configuration that yields the minimum number of total transitions;
end

Figure 1: Selection algorithm of the sub-bus.

The selection algorithm is outlined in Figure 1. As a selection metric, we use the transition probability together with average of transition correlation coefficients with the bus lines already selected (L7 of Figure 1), which is based on the observation that the maximal gain can be obtained when we invert bus lines having high probability to have transitions together.

In CMOS circuits, the dynamic power is proportional to load capacitance and switching activity. Based on this property, we define the *effective total bus transitions*, denoted by T_{eff}, as follows:

$$T_{eff} = T_{bus} + \frac{C_{int}}{C_{bus}} \cdot T_{int}, \quad (4)$$

where T_{bus} is the total bus transitions, T_{int} is the total number of transitions in the encoding/decoding circuits, C_{int} is the average capacitance of the node in the internal circuits, and C_{bus} is the total off chip capacitance. By using the equation (4), we count the number of effective transitions at L10 of Figure 1 to include effects of the encoding/decoding circuit. While we can obtain the value of T_{bus} by simply counting the number of transitions from the encoded patterns, it is not easy to obtain the accurate value of T_{int}. However, such accuracy is not needed for our purpose because T_{int} is multiplied by a relatively small constant before it is added to T_{bus}. We take a probabilistic approach to estimate T_{int}, which is given by $T_{int} = \kappa N(m+1) a_p L$, where κ denotes gate equivalents of a full adder, a_p is the average transition probability of m bus lines, and L is the number of patterns. $N(x)$ is the number of full adders used in the majority voter with x inputs and is approximately equal to $x - 2$.

4 Experimental Results

In this section we examine the efficiency of the PBI coding with two experiments. The first one is for data address patterns in benchmark examples collected from typical image or signal processing algorithms. The second one is also for the data address patterns in the example of an *audio decoder*, which is designed with VHDL and then synthesized with the LSI 10k gate library. For the effective total bus transitions, we assume 30 pF for C_{bus}, 0.2 pF for C_{int}, and 7 for κ.

4.1 Benchmark Examples

We experiment with several benchmark programs [5] that are usually implemented as a part of a system consisting of

Table 1: Comparison of the total bus transitions for benchmark examples

Applications		Unencoded	BI coding		Heuristic+PBI coding			SA+PBI coding		
Name	n	T_{bus}	T_{bus}	% Red.	T_{bus}	% Red.	$\|S\|$	T_{bus}	% Red.	$\|S\|$
Compress	32	1756468	1066266	39.3%	722260	58.9%	20	721864	58.9%	20
Laplace	32	3928218	2377233	39.5%	1603476	59.2%	19	1603470	59.2%	19
Linear	32	3948001	2420801	38.7%	1227401	68.9%	23	1227401	68.9%	23
Lowpass	32	1101927	656119	40.5%	399686	63.7%	18	399622	63.7%	20
SOR	32	2874978	1900735	33.9%	1343694	53.3%	18	1343654	53.3%	19
Wavelet	32	2197	1394	36.6%	620	71.8%	22	617	71.9%	21
Average				38.1%		62.6%			62.7%	

Table 2: Comparison of the effective total bus transitions for benchmark examples

Applications		Unencoded	BI coding		Heuristic+PBI coding			SA+PBI coding		
Name	n	T_{eff}	T_{eff}	% Red.	T_{eff}	% Red.	$\|S\|$	T_{eff}	% Red.	$\|S\|$
Compress	32	1756468	1145673	34.8%	777984	55.7%	16	773465	56.0%	15
Laplace	32	3928218	2554821	35.0%	1726454	56.0%	16	1716934	56.3%	15
Linear	32	3948001	2599283	34.2%	1395489	64.7%	23	1395489	64.7%	23
Lowpass	32	1101927	705935	35.9%	436525	60.4%	17	433850	60.6%	16
SOR	32	2874978	2030708	29.4%	1433641	50.1%	16	1433641	50.1%	16
Wavelet	32	2197	1493	32.0%	709	67.7%	21	697	68.3%	19
Average				33.6%		59.1%			59.3%	

ASICs and off chip memories connected by buses. We assume 32-b wide data address buses for all the programs. For each application, we first extract the data address patterns of memory accesses generated by a SPARC processor. Then we obtain the results after running the proposed algorithm, which is summarized in the fourth column of Table 1. The reduction of bus transitions with the PBI coding is 62.6% on the average and up to 71.8% compared to unencoded patterns. The last column in Table 1 corresponds to the PBI coding after bus lines are selected using simulated annealing instead of the heuristic algorithm, which is to have an idea of how good the solutions obtained by the proposed heuristic algorithm are. Table 2 shows the results including the effect of internal transitions of encoding/decoding circuits obtained by equation (4). The difference of reduction of bus transitions between the PBI coding and the BI coding is 25.5%, which is larger than 24.5% of Table 1. Note that the number of bus lines selected for the PBI coding is reduced further compared to that in Table 1.

4.2 Examples from MPEG-2 Audio and AC-3 Decoder

We experiment with two sets of data address patterns extracted from an audio decoder, which can support MPEG-2 audio and AC-3 standard with programmability [6]. The first one is from a *Parser processor*, which reads input data stored in a frame memory and uses data address of 16-b wide to access the external memory. The second one is from a *FFT processor*, which accesses memory via data address of 7-b wide for 128-point complex FFT. The effective total bus transitions obtained by the BI coding and the PBI coding are shown in Table 3 with the first set of patterns named *Parser* and the second set of patterns named *FFT*. The BI coding has little chance to invert patterns because the Hamming distance between two consecutive patterns is not larger than $n/2$ for the most cases for the two sets of patterns. This explains the negative reduction of the BI coding in Table 3. Note also that we are considering the overhead due to the encoding/decoding circuits. However, the power reduction with the PBI coding is still substantial in these examples.

5 Conclusion

In this paper, we propose a new bus coding scheme, which reduces the number of bus transitions for low power applications. In the proposed scheme, we minimize the number of bus lines involved in bus encoding as well as the number of total bus transitions. We present a heuristic algorithm to select a sub-group of bus lines such that bus transitions are minimized by encoding only those bus lines. The coding scheme is particularly suitable for memory-intensive special-purpose applications. However, the scheme is general enough to be used in other types of buses. Experimental results show that we can reduce the number of bus transitions substantially for benchmark examples and for a real design example. The performance of the proposed heuristic is compared to that of simulated annealing, which shows that the heuristic is highly effective.

Table 3: Comparison of the effective total bus transitions for audio decoder

Applications		Unencoded	BI coding		PBI coding		
Name	n	T_{eff}	T_{eff}	% Red.	T_{eff}	% Red.	$\|S\|$
Parser	16	2563	2675	-4.4%	741	71.1%	7
FFT	7	1036	1049	-1.3%	829	19.9%	2

Acknowledgment

The authors would like to thank Seokjun Lee and Prof. Wonyong Sung of Seoul National University for providing us example patterns of an audio decoder [6].

References

[1] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 663–670, June 1994.

[2] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. on VLSI Systems*, vol. 3, pp. 49–58, Mar. 1995.

[3] L. Benini, G. D. Micheli, E. Macii, D. Sciuto, and C. Silvano, "Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems," in *Proc. Great Lakes Symposium on VLSI*, pp. 77–82, Mar. 1997.

[4] L. Benini, G. D. Micheli, E. Macii, M. Poncino, and S. Quer, "System-level power optimization of special purpose applications: The Beach Solution," in *Proc. Int'l Symposium on Low Power Electronics and Design*, pp. 24–29, Aug. 1997.

[5] P. Panda and N. Dutt, "1995 high level synthesis design repository," in *Proc. Int'l Symposium on System Synthesis*, 1995.

[6] S. Lee and W. Sung, "A parser processor for MPEG-2 audio and AC-3 decoding," in *Proc. Int'l Symposium on Circuits and Systems*, pp. 2621–2624, June 1997.