Novel low-power bus invert coding methods with crosstalk detector

Chia-Hao Fang and Chih-Peng Fan*

Department of Electrical Engineering, National Chung Hsing University, 250 Kuo Kuang Road, Taichung 402, Taiwan, R.O.C.

(Received 24 February 2009; final version received 26 October 2009)

In System-on-a-Chip designs, crosstalk effects cause serious problems, such as wire propagation delay, noise, and power dissipation. In this article, we propose two new bus coding methods to reduce the dynamic power dissipation and wire propagation delay on buses efficiently. The proposed methods reduce both the dynamic power dissipation and wire propagation delay more than the existing bus coding methods do. Experimental results show that the crosstalk detector bus invert (CDBI) method reduces coupling activity to 25.7% from 36.4% and switching activity to 4.5% from 8.5% on 8-bit to 32-bit data buses. It reduces total power dissipation more than the other bus coding methods when a load capacitance is more than 0.3 pF/bit with UMC 0.09-μm CMOS technology. The enhanced CDBI (ECDBI) coding method reduces coupling activity to 28.4% from 38.4% and switching activity to 10.1% from 14% on 8-bit to 32-bit data buses. It reduces total power dissipation more than the other bus coding methods when a load capacitance is more than 0.2 pF/bit with UMC 0.09-μm CMOS technology. For a 0.8 pF/bit load capacitance, both the proposed methods reduce total power use by 19.3–30.9% when systems are implemented with UMC 0.09-μm CMOS technology. Similarly, both the proposed methods also reduce total power consumption more than the other bus coding methods with TSMC 0.18-μm CMOS technology. Meanwhile, the CDBI and the ECDBI schemes reduce total propagation delay up to 31.8% and 34.2%, respectively, on 32-bit data buses.

Keywords: bus coding; low-power; coupling activity; switching activity; system on a chip

1. Introduction

Power dissipation is one of the most important design specifications for low-power System-on-a-Chip (SoC) designs. Load capacitance ($C_L$) and coupling capacitance ($C_C$) are two major sources of dynamic power dissipation and wire propagation delay. If coupling capacitance increases, dynamic power consumption and wire propagation delay on a bus will also increase. Coupling capacitance also depends upon data-dependent transitions, and the increase or decrease of crosstalk effects is dependent on the relative switching between adjacent bus wires (Vittal and Marek-Sadowska 1997, Macchiarulo et al. 2002, Kaul et al. 2004, Ghoneima et al. 2006, Khan et al. 2006, Lyuh and Kim 2006). Thus, the overall performance of the system may be degraded. Consequently, the crosstalk effect is an important factor during the design process.

There are many approaches to reducing inter-wire capacitances (Cong 2001, Elgamel and Bayoumi 2003). The first approach applies bus coding schemes to reduce the dynamic power consumption. It mainly reduces switching activity on a bus. Such coding methods are applied to two different bus modes, i.e., data buses and address buses. In the previous well-known bus coding techniques (Micea and Wayne 1995, Benini et al. 1997, Youngsoo et al. 1998, 2001a, 2001b, Ramprasad et al. 1999, Fornaciari et al. 2000), the INC-XOR (Ramprasad et al. 1999), T0 (Benini et al. 1997), and T0-XOR (Fornaciari et al. 2000) coding schemes are designed for instruction address buses because the instruction address is predictive. The hihrTS (Youngsoo et al. 2001a), bus-invert (BI) (Micea and Wayne 1995), and partial bus-invert (PBI) (Youngsoo et al. 1998, 2001b) coding schemes are designed for data buses which are generally random in values. In Micea and Wayne (1995), the number of transmitting transitions does not exceed half of the bus width. When the number of transmitting transitions is more than half of the bus width, the output word is inverted and the control line is set to ‘High.’ Otherwise, the original data are transmitted and the control line is

*Corresponding author. Email: d9464305@mail.nchu.edu.tw; cpfan@dragon.nchu.edu.tw

ISSN 0253–3839 print/ISSN 2158–7299 online
© 2011 The Chinese Institute of Engineers
DOI: 10.1080/02533839.2011.553030
http://www.informaworld.com
set to ‘Low.’ The PBI method is an extension of the BI method, which then partitions bus lines into two parts, and each part is coded by the BI method. It enhances the advantage of the BI method, but it requires redundant control lines, but, crosstalk effects are not considered in Ramprasad et al. (1999), Benini et al. (1997), Fornaciari et al. (2000), Youngsoo et al. (2001a), Micea and Wayne (1995), and Youngsoo et al. (1998, 2001b). In recent years, system engineers have aimed at the bus coding methods for crosstalk effect reduction. These methods are separated into two modes, including 2- and 3-bit conditions. For a 2-bit bus, the methods in Ayoub and Orailoglu (2005), Lyuh and Kim (2006), Huang and Hwang (2004), Kim et al. (2000), Macchiarulo et al. (2001), and Zhan et al. (2002) are applied to encode data buses. Similarly, there are many well-known bus coding technologies (Hirose and Yasuura 2000, Macchiarulo et al. 2002, Khan et al. 2006, Tu et al. 2006) applied to 3-bit buses. For example, Hirose and Yasuura (2000) apply repeater technology to reduce total bus delay. Khan et al. (2006) use the alternate complement value and shield wires to reduce crosstalk effects. Macchiarulo et al. (2002) use wire placement to reduce crosstalk effects and it aims at address buses.

The second approach widens the pitch between bus lines, but the total area of bus systems may grow too large. The third approach uses place and route tools to avoid the routing of bus lines side by side. In SoC systems, the interconnect complexity and routing time cause the difficulty in minimizing coupling capacitances. The fourth approach insert a shielding line (\(V_{DD}/\text{Ground}\)) between two adjacent signal lines (Vittal and Marek-Sadowska 1997, Kaul et al. 2004, Ghoneima et al. 2006). Thus, the fourth technology induces a larger bus width. In SoC systems, the crosstalk effect is an important issue to influence dynamic power dissipation and wire propagation delay (Lajolo 2001, Huang and Hwang 2004). Yu and Lin (2006) develop a comprehensive study on the viability of on-chip bus encoding methods from the perspectives of energy, delay, and peak noise reduction. Lin (2008) derives a theoretical analysis of the BI coding for coupling reduction. Lin’s discoveries include a set of closed-form formulas to compute the number of couplings per bus transfer for a non-partitioned bus versus a partitioned bus.

In this article, we mainly aim at the reduction of crosstalk effects and transition activity on a bus. Thus, the bus coding method is applied to reduce the dynamic power consumption and wire propagation delay in signal transitions. Although these coding methods in Lyuh and Kim (2006), Khan et al. (2006), Macchiarulo et al. (2002), Huang and Hwang (2004), Kim (2000), Macchiarulo et al. (2001), Duan et al. (2001), and Zhan et al. (2002) focus on the reduction of crosstalk effects, these techniques have many shortcomings. For instance, they must increase redundant control lines and hardware areas, and cannot reduce coupling and switching activities at the same time. Therefore, we present two new low-power bus coding methods to enhance previous coding methods. First, our proposed methods are not complex. Second, they simultaneously reduce coupling and switching activities. Third, the proposed methods use a smaller number of redundant shielding lines to obtain the same efficiency as the previous methods. Finally, their coding efficiencies are better than the other coding methods. By testing various random data streams, the experimental results show that the first proposed coding scheme reduces coupling activity by 25.7–36.4% and switching activity by 4.5–8.5%. It reduces total power dissipation more than the other bus coding methods when the load capacitance is more than 0.3 pF/bit with UMC 0.09-\(\mu\)m CMOS technology. Moreover, the second proposed coding scheme reduces the coupling activity by 28.4–38.4% and switching activity by 10.1–14%. It reduces total power dissipation more than the other bus coding methods when the load capacitance is more than 0.2 pF/bit with UMC 0.09-\(\mu\)m CMOS technology. For a 0.8 pF/bit load capacitance, both the proposed methods reduce total power consumption by 19.3–30.9% when systems are implemented with UMC 0.09-\(\mu\)m CMOS technology. Similarly, both the proposed methods also reduce total power consumption more than the other bus coding methods with TSMC 0.18-\(\mu\)m CMOS technology. Meanwhile, the crosstalk detector bus invert (CDBI) and the enhanced CDBI (ECDBI) schemes reduce total propagation delay up to 31.8% and 34.2%, respectively, on a 32-bit data bus.

The rest of this article is described as follows. In Section 2, we review and define the power expression, the bus models, and the dynamic power consumption of three adjacent bus wires. In Section 3, we present two new bus coding techniques and describe their advantages, and the proposed methods greatly decrease dynamic power dissipation and wire propagation delay. The simulation and implementation results and comparisons of different bus coding methods are shown in Section 4. Finally, we state a conclusion.
2. Power expression and bus models for deep submicron buses

After the bus coding skill is applied to encode the bus data, the dynamic power consumption on a 3-bit bus with the encoded data is calculated as follows:

\[
P_{D,\text{coded}} = (T_S + \lambda \times T_C) \times V_{DD}^2 \times f
\]

where \(C_L\) is a load capacitance, \(C_C\) a coupling capacitance, \(V_{DD}\) a supplying voltage, \(f\) a clock frequency, and \(\lambda\) a capacitance ratio which is defined as follows.

\[
\lambda = \frac{C_C}{C_L}.
\]

The \(\lambda\) value is dependent on technologies. For example, the interconnect width, the pitch, the aspect ratio, and the dielectric thickness will affect the \(\lambda\) value. In Equation (1), \(T_S\) indicates the average values of switching activity for load capacitance, and then \(T_C\) the average values of coupling activity for coupling capacitance. However, \(T_S\) value lies between 0 and 1. The un-coded \(T_S\) value is equivalent to 1. \(T_C\) value also lies between 0 and 1. Similarly, the un-coded \(T_C\) value equals 1. The activity defines the variation among three adjacent bit lines. The signal transitions on 3-bit lines are classified into five types, as shown in Table 1. We apply Equation (2) as follows (Hirose and Yasuura 2000):

\[
C_{\text{eff}} = C_C \times \frac{\Delta V_2 - \Delta V_1}{E} + C_C \times \frac{\Delta V_2 - \Delta V_3}{E},
\]

where \(\Delta V_2\) is the voltage variation of the center wire, \(\Delta V_1\) and \(\Delta V_3\) the voltage variations of the adjacent wires, \(E\) the power supply voltage, equaling the rail-to-rail signal voltage in CMOS circuits, and \(C_{\text{eff}}\) the coupling capacitance. For example, the Type-1 crosstalk capacitance equals \(C_C\), the Type-2 crosstalk capacitance equals \(2C_C\), etc. By the above-mentioned definition, the dynamic power consumption on un-coded buses is defined as follows:

\[
P_{D,\text{un-coded}} = (1 + \lambda) \times C_L \times V_{DD}^2 \times f.
\]

Equation (4) gives the power reduction ratio as follows.

\[
\text{Power reduction ratio} = \frac{P_{D,\text{un-coded}} - P_{D,\text{coded}}}{P_{D,\text{un-coded}}} \times 100%.
\]

Table 1. Crosstalk types of a 3-bit bus when considering RC effects.

<table>
<thead>
<tr>
<th>Type-0</th>
<th>Type-1</th>
<th>Type-2</th>
<th>Type-3</th>
<th>Type-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(\downarrow \uparrow \downarrow)</td>
<td>(\downarrow \uparrow \downarrow)</td>
<td>(\downarrow \uparrow \downarrow)</td>
<td>(\downarrow \uparrow \downarrow)</td>
<td>(\downarrow \uparrow \downarrow)</td>
</tr>
<tr>
<td>(\uparrow \downarrow \uparrow \downarrow)</td>
<td>(\uparrow \downarrow \uparrow \downarrow)</td>
<td>(\uparrow \downarrow \uparrow \downarrow)</td>
<td>(\uparrow \downarrow \uparrow \downarrow)</td>
<td>(\uparrow \downarrow \uparrow \downarrow)</td>
</tr>
<tr>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
</tr>
<tr>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
</tr>
<tr>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
<td>(\downarrow \downarrow \downarrow \downarrow)</td>
</tr>
</tbody>
</table>

Note: \(\uparrow\), denotes switch from ‘0’ to ‘1’; \(\downarrow\), switch from ‘1’ to ‘0’; \(–\), no transition.

\(P_{D,\text{coded}}\) in Equation (1) is the dynamic power consumption on a bus with encoded data.

Following some of the literature (Vittal and Marek-Sadowska 1997, Duan et al. 2001, Kaul et al. 2004, Lyuh and Kim 2006, Khan et al. 2006, Ghoneima et al. 2006, Tu et al. 2006), we explain the interconnecting bus characteristics of crosstalk effects by a simple case of three adjacent wires shown in Figure 1. We only consider coupling and load capacitances, and neglect parasitical capacitances. The dynamic power dissipation generated by the load capacitance is proportional to the number of signal transitions of bus lines. On the other hand, the dynamic power dissipation generated by the coupling capacitance depends on the signal transitions between coupled interconnects. There are five kinds of crosstalk effects shown in Table 1. We use Equation (2) to enumerate all kinds of serious crosstalk effects. Type-1 crosstalk produces less serious crosstalk effects than the other types of crosstalk, and Type-4 crosstalk produces the worst crosstalk effects of all crosstalk types. Thus, we use the Type-1 crosstalk effect as a standard to define the other crosstalk effects. Serious crosstalk effects include Type-2, Type-3, and Type-4 crosstalk.

3. Proposed low power bus coding methods

In Section 2, different kinds of crosstalk effects are described. Then, we present two new low-complexity bus coding techniques specially devised for three adjacent wires. The proposed methods can not only improve system reliability but also reduce total power consumption and total propagation delay.
3.1. CDBI method

Now we propose the first crosstalk detector coding method, called the CDBI method, and it aims at crosstalk effects to reduce dynamic power dissipation and propagation delay by reducing coupling and switching activities. The *n*-bit bus is separated into clusters of 4-bit data. Each 4-bit cluster is, respectively, encoded according to the CDBI method by the relationship of the original data and last encoding data (i.e., \( b(t) \) and \( B(t-1) \)), and then an extra control bit (i.e., the INV line) is obtained, and the unique decoding ability is achieved.

Figure 2 shows 4-bit bus encoder circuits. The encoder circuits are composed of a crosstalk detector, a selector, and a NOT gate. First, the crosstalk detector judges whether the \( b(t) \) value (i.e., the bus value to be sent presently on the bus at time *t*) will cause crosstalk effects about the previous bus state \( B(t-1) \) (i.e., the encoded bus value to be sent last on bus lines at time \( t-1 \)). Second, the crosstalk detector generates a control signal and the INV line is set to the OC signal. Then, the encoder circuit uses the OC signal to select the encoding value. Therefore, the \( B(t) \) value does not produce the worst crosstalk effects. The internal circuits of the crosstalk detector are composed of basic logic gates shown in Figure 3(a). We combine the function of the selector circuit and a NOT gate, and then we get optimum circuits shown in Figure 3(b). Thus, the CDBI method is low complexity in implementation, and it uses the crosstalk detector for a low-power design and the minimization of the worst crosstalk effects.

The algorithm of the CDBI method is described as follows. The crosstalk detector first receives the \( b(t) \) and \( B(t-1) \) values, which are reciprocally compared. The purpose of the crosstalk detector is to observe whether adjacent wires generate transition 01 → 10 or 10 → 01. If a transition happens, a three-input AND gate may obtain the ‘High’ signal because adjacent wires produce crosstalk effects. In addition, we can group the crosstalk detector into three contrast circuits. Each contrast circuit detects crosstalk effects between wires 1 and 2, between wires 2 and 3, or between wires 3 and 4. After 3 three-input AND gates are operated, the contrast circuits produce 3-bit outputs, and these outputs pass through a three-input OR gate to acquire the OC signal, which is exported from the crosstalk detector. The purpose of a three-input OR gate is to determine whether two adjacent wires cause the transition 01 → 10 or 10 → 01. If a three-input OR gate is set to ‘High,’ then the OC signal is also set to ‘High.’ Therefore, the \( B(t) \) value (i.e., the encoded bus value to be sent on bus lines at time *t*) acquires the \( b(t) \) value and INV is set to ‘Low’ when the OC signal is set to ‘Low.’ The proposed CDBI algorithm is shown as follows:

\[
\begin{align*}
\text{if}(\text{OC} == 1) & \quad B(t) = b(t) \\
\text{else} & \quad \text{Set INV} \\
& \quad B(t) = b(t) \\
& \quad \text{Reset INV}
\end{align*}
\]

Each 4-bit encoded datum needs an additional control bit. Therefore, after the bus encoding is used, the *n*-bit bus is extended to \( n + n/4 \) bits, i.e., the *n*-bit bus encoding has a wire redundancy of 25%. The CDBI method generates two complement values (either \( b(t) \) or \( \overline{b(t)} \)), and then the OC signal is applied to choose between them, i.e., the OC signal is set to 1 if the \( b(t) \) value produces crosstalk effects; otherwise, it is set to 0. We observe that one of the two data greatly
reduces Type-4, Type-3, and Type-2 crosstalk to about the $B(t-1)$ value. In other words, the CDBI method turns Type-2, Type-3, and Type-4 crosstalk into Type-0 and Type-1 crosstalk. However, Type-1 and Type-0 crosstalk need not be considered.

### 3.2. ECDBI method

The CDBI method generates a slight amount of Type-4 crosstalk and greatly reduces Type-2 and Type-3 crosstalk from 4-bit clusters. However, it reduces switching activity slightly. For this reason, we propose the second bus coding method to improve the coding efficiency of the CDBI method. The second proposed bus coding method is called the ECDBI coding method. It aims at further reductions of crosstalk effects and switching activity. The architecture of the ECDBI method is shown in Figure 4. It includes a switching detector and a two-input OR gate, different from the CDBI method. The internal circuits of the joint crosstalk/switching detector are composed of the basic logic gates shown in Figure 5. The upper part performs the major function of the crosstalk detector, and the lower part the major function of the switching detector. The purpose of the switching detector is to reduce switching activity. The purpose of a two-input
The(Encoding) value exploits the EBW signal to acquire the \( b(t) \) value, and then INV signal is set to ‘Low.’

The circuit of the switching detector in the ECDBI method equals that in the PBI method. Therefore, the ECDBI method possesses the advantages of the CDBI and PBI methods simultaneously. Except for the switching detector circuit, the other functions of the ECDBI method equal those of the CDBI method. Therefore, we omit the discussion of the same functions.

The bus coding technology must provide both bus encoding and decoding functions; otherwise, it cannot be used for system applications. Figure 6 shows 4-bit bus decoder circuits. The decoder circuit is realized with low complexity. The internal circuits of the decoder consist of the selector and a NOT gate. The decoder receives the signals, which are the \( B(t) \) value, the \( \overline{B}(t) \) value, and the INV line. When the \( B(t) \) value is transmitted to the decoder, the decoder uses the INV signal to decode the original bus data, \( b(t) \). If the INV line is set to ‘High’ value, the \( b(t) \) value equals the \( B(t) \) value. Otherwise, when the INV line is set to ‘Low,’ the \( b(t) \) value equals the \( \overline{B}(t) \) value. In addition, the CDBI and PBI methods also use the same decoding circuit as the ECDBI method. Furthermore, we simplify the decoding circuit to obtain the optimum circuit, which is the same as the one given in Figure 3(b). The proposed decoding scheme is described as follows:

\[
\begin{align*}
\text{if (INV} &= 1) \\
B(t) &= B(t) \\
\text{else} \\
B(t) &= \overline{B}(t) \\
\text{Reset INV}
\end{align*}
\]

where the ECDBI algorithm is shown as follows:

\[
\begin{align*}
\text{if (OC} &= 1 \text{or EBW} = 1) \\
B(t) &= \overline{b}(t) \\
\text{Set INV} \\
\text{else} \\
B(t) &= b(t) \\
\text{Reset INV}
\end{align*}
\]

The ECDBI method equal those of the CDBI method. Therefore, the ECDBI method possesses the advantages of the CDBI and PBI methods simultaneously. Except for the switching detector circuit, the other functions of the ECDBI method equal those of the CDBI method. Therefore, we omit the discussion of the same functions.

The bus coding technology must provide both bus encoding and decoding functions; otherwise, it cannot be used for system applications. Figure 6 shows 4-bit bus decoder circuits. The decoder circuit is realized with low complexity. The internal circuits of the decoder consist of the selector and a NOT gate. The decoder receives the signals, which are the \( B(t) \) value, the \( \overline{B}(t) \) value, and the INV line. When the \( B(t) \) value is transmitted to the decoder, the decoder uses the INV signal to decode the original bus data, \( b(t) \). If the INV line is set to ‘High’ value, the \( b(t) \) value equals the \( B(t) \) value. Otherwise, when the INV line is set to ‘Low,’ the \( b(t) \) value equals the \( \overline{B}(t) \) value. In addition, the CDBI and PBI methods also use the same decoding circuit as the ECDBI method. Furthermore, we simplify the decoding circuit to obtain the optimum circuit, which is the same as the one given in Figure 3(b). The proposed decoding scheme is described as follows:

\[
\begin{align*}
\text{if (INV} &= 1) \\
\overline{b}(t) &= B(t) \\
\text{else} \\
B(t) &= \overline{B}(t)
\end{align*}
\]

The circuit of the switching detector in the ECDBI method equals that in the PBI method. Therefore, the ECDBI method possesses the advantages of the CDBI and PBI methods simultaneously. Except for the switching detector circuit, the other functions of the ECDBI method equal those of the CDBI method. Therefore, we omit the discussion of the same functions.
and clusters when the bus is divided into \( n/4 \) clusters, and the bus is also expanded spatially to carry redundant bits. Therefore, there are some coding methods using shielding wires. Figure 7(b) shows an 8-bit bus using the method in Khan et al. (2006), which simultaneously uses coding technology and three redundant shielding lines to reduce the worst crosstalk pattern. The 8-bit bus width is extended to 13-bit bus width, i.e., \( 8 + 2 + 3 = 13 \). In Figure 7(c), the proposed methods only use a redundant shielding line to reduce the worst crosstalk pattern between inter-cluster regions. It is noted that the interconnect routing

Figure 5. The joint circuit of the crosstalk/switching detector with a 1-bit output.

Figure 6. The decoder architecture for a generic 4-bit bus.
scheme in Figure 7(c) is applied to all compared coding methods in this article for equitable comparisons. We find that a 32-bit width bus can be encoded with 47 wires by our proposed methods, compared with 55 wires as mentioned by existing papers (Khan et al. 2006, Lyuh and Kim 2006). For this reason, our proposed coding methods use a smaller number of redundant shielding lines to obtain the same efficiency as the previous methods.

4. Experimental results and comparisons

In this section, we evaluate and compare all kinds of bus coding performance, such as switching activity, coupling activity, areas, power consumption, and the total propagation delay of the codec. To simulate different data bus coding techniques, we adopt 8-bit, 16-bit, and 32-bit data bus lines to compare their dynamic power dissipation and wire propagation delay. For tests, the test steams include random data, image files, PPT files, MP3 files, and PDF files.

4.1. Power reduction results and comparisons

In the simulation with random patterns, the BI method performs at better self-switching reductions because it uses an extra control line to minimize switching activity, and then the dynamic power dissipation is reduced greatly. Simultaneously, it also slightly reduces coupling activity. According to the papers in Ayoub and Orailoglu (2005), Victor and Keutzer (2001), Hirose and Yasuura (2000), Lyuh and Kim (2006), Khan et al. (2006), Vittal and Marek-Sadowska (1997), Ghoneima et al. (2006), Duan et al. (2001), Zhan et al. (2002), Tu et al. (2006), Cong (2001), and Elgamel and Bayoumi (2003), coupling activity is more serious than switching activity. To reduce dynamic power dissipations, the PBI method is better than the BI coding method because it uses smaller bus segmentation to increase coding efficiency. However, it needs a lot of extra lines. For instance, the PBI method in an 8-bit bus calls for two extra control lines. The simple-odd/even bus invert (OEBI) method (Zhan et al. 2002) is equivalent to the PBI method. Therefore, it has the same efficiency and bus lines as the PBI method. The calculated-OEBI method (Zhan et al. 2002) uses complex circuits to reduce the worst crosstalk pattern. Although it can greatly reduce coupling activity and slightly reduce switching activity at the same time, it increases hardware cost a lot. The method in Duan et al. (2001) uses a codebook to reduce the worst crosstalk pattern. Although it can eliminate the Type-4 and Type-3 crosstalk, it greatly increases the Type-0, Type-1, and Type-2 crosstalk. Besides, it also results in a
large number of switching activities. The method in Khan et al. (2006) greatly reduces coupling activity and slightly reduces switching activity at the same time because it mainly aims at crosstalk effects. Therefore, the methods in Khan et al. (2006), Youngsoo et al. (1998, 2001b), Duan et al. (2001), and Zhan et al. (2002) reduce dynamic power dissipation more than the previous coding methods do. The coding methods in Khan et al. (2006), Youngsoo et al. (1998, 2001b), Duan et al. (2001), Zhan et al. (2002), and our proposed methods use the same bus lines. In other words, the original data bus is changed from 8-bit to 11-bit width, from 16-bit to 23-bit width, or from 32-bit to 47-bit width.

We calculate the worst crosstalk couplings and switching activities from data on buses at the same time. The different bus coding efficiencies are shown in Figures 8 and 9. Figure 8 shows the reduction of coupling and switching activities in 8-bit image data, and Figure 9 shows the reduction of coupling and switching activities in 8-bit random data. In Figure 8, the PBI method reduces more switching activity than the other coding methods do. To reduce Type-2 crosstalk, the method in Duan et al. (2001) is worse than the other methods. To reduce Type-3 crosstalk, the methods in Khan et al. (2006), Duan et al. (2001), Zhan et al. (2002), and our proposed methods are better than the other coding methods. To reduce Type-4 crosstalk, the methods in Micea and Wayne (1995), Kim et al. (2000) and Zhan et al. (2002) are worse than the other methods. To produce Type-1 crosstalk, the proposed CDBI method and the methods in Youngsoo et al. (1998, 2001b), Duan et al. (2001), and Zhan et al. (2002) are better than the other methods. Although Type-0 crosstalk is greatly increased, the occurrence does not cause dynamic power dissipation. The Type-0 and Type-1 crosstalk have been mentioned in Section 2. In Figure 9, all cases of the coding methods are similar to the above-mentioned results in Figure 8.

In Figure 8, the proposed CDBI method reduces Type-2 crosstalk by 59%, Type-3 by 73.8%, and Type-4 by 95%, but Type-1 and Type-0 are increased by 31% and 13.5%, respectively. The reduction of $T_S$ is 7.8% and that of $T_C$ is 25.9%. Furthermore, the proposed ECDBI method reduces Type-2 crosstalk by 69.5%, Type-3 by 72.1%, and Type-4 by 95.2%, but the Type-1 and Type-0 are increased by 28.9% and 51.4%, respectively. The reduction of $T_S$ is 14% and that of $T_C$ is 28.5%. Thus, the proposed methods efficiently reduce coupling and switching activities at the same time. By testing various random data streams, the experimental results show that the proposed CDBI method reduces coupling activity by 25.7–36.4% and switching activity by 4.5–8.5%. Moreover, the ECDBI method reduces coupling activity by 28.4–38.4% and switching activity by 10.1–14%.

According to the 8-bit statistics in Figures 8 and 9, the comparisons in the 16-bit and 32-bit bus modes have the same coding efficiencies as shown in Figures 8 and 9; so, the comparisons in 16-bit and 32-bit bus modes are omitted.
Table 2 shows the comparison of dynamic power parameters in the coupling and switching activities with various bit widths for each benchmark. The sizes of the test files, i.e., random data, image files, PPT files, MP3 files, and PDF files, are 4.78, 1.63, 8.1, 2.78, and 0.63 MB, respectively. To compare our methods with the un-coded data bus, we simulate five different test files (i.e., the random data, image files, PPT files, MP3 files, and PDF files) to calculate the averages of coupling activity and switching activity, and the $\lambda$ parameter. The $\lambda$ parameter is set to 10, which is in agreement with previous papers (Vittal and Marek-Sadowska 1997, Hirose and Yasuura 2000, Duan et al. 2001, Victor and Keutzer 2001, Ayoub and Orailoglu 2005, Ghoneima et al. 2006, Khan et al. 2006, Lyuh and Kim 2006). In this article, the $\lambda$ parameter is set to 10, which is in agreement with previous papers.
parameter ranges from 1 to 5. By Equation (1), the dynamic power dissipation from coupling activities is larger than that from switching activities when the \( \lambda / C_{21} \) parameter becomes larger. The method in Micea and Wayne (1995) is an exception because it mainly aims at switching activities. The proposed CDBI and the methods in Khan et al. (2006), Youngsoo et al. (1998, 2001), Kim et al. (2000), Duan et al. (2001) and Zhan et al. (2002) are worse than the methods in Youngsoo et al. (1998, 2001b) when the \( \lambda / C_{21} \) parameter is 1. The coding efficiency of the proposed CDBI method is equivalent to that of the method in Khan et al. (2006). The coding efficiency is less than 1% when the proposed CDBI method is compared with the method in Khan et al. (2006). The CDBI method is better than the methods in Micea and Wayne (1995), Youngsoo et al. (1998, 2001b), Kim et al. (2000), Duan et al. (2001) and Zhan et al. (2002) when the \( \lambda / C_{21} \) parameter exceeds 2. Thus, it reduces coupling and switching activities by from 20.3% to 34.1% in comparison with the un-coded method.

To reduce coupling and switching activities, the proposed ECDBI method performs with better coding efficiency than the other methods with various bit widths for each benchmark. It reduces coupling and switching activities by from 23.8% to 36.4%, compared with the un-coded method. Moreover, it reduces coupling and switching activities by from 1.1% to 20.9%, compared with other coding methods, and it also improves the efficiency of the CDBI method by from 2.3% to 3.5% shown in Table 2.

Every data bus coding method may incur some overheads, including area and power. In Table 3, we estimate the actual overheads of different bus encoder and decoder circuits with various bit widths. We use the Verilog HDL to model and simulate different bus encoder and decoder circuits, and we also simplify all coding hardware to obtain the optimum circuits. Then, we use the Design Vision™ logic synthesizer with TSMC CMOS 0.18-\( \mu \)m and UMC 0.09-\( \mu \)m standard cell technologies to evaluate the power consumption and the area of bus codec circuits. Furthermore, the operation frequency of different bus codecs is set to 125 MHz, and the operating voltages are 1.8 and 1 Volts for the 0.18 and 0.09-\( \mu \)m technologies, respectively. By reading Table 3, the chip area of the calculated-OEBI circuit is larger than the other coding methods. Thus, it causes the most serious hardware overheads. The proposed CDBI method has the same capability as the method in Khan et al. (2006) but the hardware cost of the CDBI method is less than that of the method in Khan et al. (2006). The area of the ECDBI codec is a little larger than that of the CDBI codec. These coding methods with various bit widths are selected for benchmarks, and their power reduction efficiencies for both coupling and switching activities are shown in Figure 10, by setting the \( \lambda / C_{21} \) parameter to three for estimations of dynamic power

<table>
<thead>
<tr>
<th>Technology (( \mu )m)</th>
<th>Coding methods</th>
<th>8-Bit</th>
<th>16-Bit</th>
<th>32-Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSMC 0.18</td>
<td>CBI (Kim et al. 2000)</td>
<td>2917/0.65</td>
<td>6034/1.32</td>
<td>13768/2.48</td>
</tr>
<tr>
<td></td>
<td>BI (Micea and Wayne 1995)</td>
<td>3109/0.7</td>
<td>6258/1.45</td>
<td>14516/2.57</td>
</tr>
<tr>
<td></td>
<td>PBI (Youngsoo et al. 1998, 2001)</td>
<td>2831/0.6</td>
<td>5782/1.21</td>
<td>12764/2.31</td>
</tr>
<tr>
<td></td>
<td>Simple-OEBI (Zhan et al. 2002)</td>
<td>2831/0.6</td>
<td>5782/1.21</td>
<td>12764/2.31</td>
</tr>
<tr>
<td></td>
<td>Calculated-OEBI (Zhan et al. 2002)</td>
<td>14314/1.37</td>
<td>28668/3.99</td>
<td>62879/7.57</td>
</tr>
<tr>
<td></td>
<td>Khan et al. (2006)</td>
<td>3250/0.73</td>
<td>6523/1.59</td>
<td>13546/2.45</td>
</tr>
<tr>
<td></td>
<td>Duan et al. (2001)</td>
<td>2318/0.58</td>
<td>4890/1.01</td>
<td>10938/2.21</td>
</tr>
<tr>
<td></td>
<td>CDBI</td>
<td>2860/0.64</td>
<td>5826/1.28</td>
<td>12752/2.31</td>
</tr>
<tr>
<td></td>
<td>ECDBI</td>
<td>2934/0.65</td>
<td>5978/1.29</td>
<td>13056/2.35</td>
</tr>
<tr>
<td>UMC 0.09</td>
<td>CBI (Kim et al. 2000)</td>
<td>1685/0.15</td>
<td>3420/0.31</td>
<td>6935/0.6</td>
</tr>
<tr>
<td></td>
<td>BI (Micea and Wayne, 1995)</td>
<td>1705/0.16</td>
<td>3512/0.32</td>
<td>7097/0.61</td>
</tr>
<tr>
<td></td>
<td>PBI (Youngsoo et al. 1998, 2001b)</td>
<td>1596/0.14</td>
<td>3216/0.27</td>
<td>6632/0.57</td>
</tr>
<tr>
<td></td>
<td>Simple-OEBI (Zhan et al. 2002)</td>
<td>1596/0.14</td>
<td>3216/0.27</td>
<td>6632/0.57</td>
</tr>
<tr>
<td></td>
<td>Calculated-OEBI (Zhan et al. 2002)</td>
<td>4661/0.38</td>
<td>9285/0.77</td>
<td>18645/1.95</td>
</tr>
<tr>
<td></td>
<td>Khan et al. (2006)</td>
<td>1850/0.19</td>
<td>3650/0.35</td>
<td>7175/0.65</td>
</tr>
<tr>
<td></td>
<td>Duan et al. (2001)</td>
<td>1392/0.13</td>
<td>2830/0.26</td>
<td>5679/0.52</td>
</tr>
<tr>
<td></td>
<td>CDBI</td>
<td>1632/0.14</td>
<td>3288/0.28</td>
<td>6721/0.59</td>
</tr>
<tr>
<td></td>
<td>ECDBI</td>
<td>1658/0.14</td>
<td>3322/0.28</td>
<td>6789/0.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area (( \mu )m(^2))/Power (mW) for bus encoder + bus decoder</th>
<th>8-Bit</th>
<th>16-Bit</th>
<th>32-Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology (( \mu )m) Coding methods</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSMC 0.18</td>
<td>CBI (Kim et al. 2000)</td>
<td>2917/0.65</td>
<td>6034/1.32</td>
</tr>
<tr>
<td>BI (Micea and Wayne 1995)</td>
<td>3109/0.7</td>
<td>6258/1.45</td>
<td>14516/2.57</td>
</tr>
<tr>
<td>PBI (Youngsoo et al. 1998, 2001)</td>
<td>2831/0.6</td>
<td>5782/1.21</td>
<td>12764/2.31</td>
</tr>
<tr>
<td>Simple-OEBI (Zhan et al. 2002)</td>
<td>2831/0.6</td>
<td>5782/1.21</td>
<td>12764/2.31</td>
</tr>
<tr>
<td>Calculated-OEBI (Zhan et al. 2002)</td>
<td>14314/1.37</td>
<td>28668/3.99</td>
<td>62879/7.57</td>
</tr>
<tr>
<td>Khan et al. (2006)</td>
<td>3250/0.73</td>
<td>6523/1.59</td>
<td>13546/2.45</td>
</tr>
<tr>
<td>Duan et al. (2001)</td>
<td>2318/0.58</td>
<td>4890/1.01</td>
<td>10938/2.21</td>
</tr>
<tr>
<td>CDBI</td>
<td>2860/0.64</td>
<td>5826/1.28</td>
<td>12752/2.31</td>
</tr>
<tr>
<td>ECDBI</td>
<td>2934/0.65</td>
<td>5978/1.29</td>
<td>13056/2.35</td>
</tr>
<tr>
<td>UMC 0.09</td>
<td>CBI (Kim et al. 2000)</td>
<td>1685/0.15</td>
<td>3420/0.31</td>
</tr>
<tr>
<td>BI (Micea and Wayne, 1995)</td>
<td>1705/0.16</td>
<td>3512/0.32</td>
<td>7097/0.61</td>
</tr>
<tr>
<td>PBI (Youngsoo et al. 1998, 2001b)</td>
<td>1596/0.14</td>
<td>3216/0.27</td>
<td>6632/0.57</td>
</tr>
<tr>
<td>Simple-OEBI (Zhan et al. 2002)</td>
<td>1596/0.14</td>
<td>3216/0.27</td>
<td>6632/0.57</td>
</tr>
<tr>
<td>Calculated-OEBI (Zhan et al. 2002)</td>
<td>4661/0.38</td>
<td>9285/0.77</td>
<td>18645/1.95</td>
</tr>
<tr>
<td>Khan et al. (2006)</td>
<td>1850/0.19</td>
<td>3650/0.35</td>
<td>7175/0.65</td>
</tr>
<tr>
<td>Duan et al. (2001)</td>
<td>1392/0.13</td>
<td>2830/0.26</td>
<td>5679/0.52</td>
</tr>
<tr>
<td>CDBI</td>
<td>1632/0.14</td>
<td>3288/0.28</td>
<td>6721/0.59</td>
</tr>
<tr>
<td>ECDBI</td>
<td>1658/0.14</td>
<td>3322/0.28</td>
<td>6789/0.6</td>
</tr>
</tbody>
</table>
dissipation. We use Equation (3) as the un-coded base power to evaluate the percentage of power reduction efficiency with various bit widths by each benchmark, and we also use Equations (1), (3), and (4) to calculate the reduction of power dissipation. The proposed CDBI method reduces dynamic power consumption more than the methods in Micea and Wayne (1995), Kim et al. (2000) and Zhan et al. (2002) do, but it is not better than the methods in Khan et al. (2006), Youngsoo et al. (1998, 2001b), and Zhan et al. (2002). It reduces dynamic power consumption by from 21% to 29.2% with various bit widths for each benchmark. Then, the proposed ECDBI method reduces dynamic power consumption more than the other methods in each benchmark. It also reduces dynamic power consumption by from 24.6% to 32.2% with various bit widths for each benchmark. Thus, the ECDBI method reduces dynamic power consumption by 1.5–19.4% when compared with the other coding methods. The power consumption of the ECDBI codec is also a little larger than that of the CDBI codec.

The power consumption at the bus encoder is defined as $P_{\text{enc}}$, and that at the bus decoder is defined as $P_{\text{dec}}$. In order to estimate the total power dissipation, the additional power dissipation on the bus encoder and decoder sides must be considered. In Equation (1), $P_{\text{D,coded}}$ is generated by the reduced switching and the coupling activities after bus coding methods are used. For various widths, Figure 11 shows the comparisons of different bus coding methods for the total power ratio (including $P_{\text{enc}}$, $P_{\text{D,coded}}$, and $P_{\text{dec}}$) versus a load capacitance per bit line with TSMC 0.18-$\mu$m CMOS technology. When the 0.18-$\mu$m technology is applied, the CBI, BI, PBI, simple-OEBI, calculated-OEBI, Khan and Duan coding methods reduce total power consumption by 6.4–8.8%, 11.1–11.2%, 17.5–27.1%, 15.8–25.7%, 11.5–14.7%, 17–25.5%, and 15.3–25.4% with various bit widths at a 0.8 pF/bit load capacitance, respectively. The proposed CDBI and ECDBI methods reduce total power consumption by 17.5–26.6% and 20.4–29.1% with various bit widths at a 0.8 pF/bit load capacitance, respectively.

Figure 12 shows that all coding methods require small power consumption when the technology is scaled down. When 0.09-$\mu$m technology is applied, the power consumption of the bus encoder and decoder shown in Table 3 become small, but $P_{\text{D,coded}}$ becomes a burden. With 0.09-$\mu$m technology, the CBI, BI, PBI, simple-OEBI, calculated-OEBI, Khan and Duan coding methods reduce total power use by...
7.9–10.3%, 13.1–13.3%, 19.1–28.7%, 17.4–27.3%, 12.8–21.9%, 18.2–27.7%, and 16.9–26.2% with various bit widths at a 0.8 pF/bit load capacitance, respectively. The proposed CDBI and ECDBI methods reduce total power use by 19.3–28.5% and 22.2–30.9% with various bit widths at a 0.8 pF/bit load capacitance, respectively.

4.2. Delay reduction results and comparisons

Section 4.1 discusses the improvement of power reduction by diminishing the worst crosstalk effects. Since the proposed bus codec is inserted, the performance influence includes the wire propagation delay reduction by diminishing the worst crosstalk effects and the extra circuit delays generated by the encoder/
decoder. In Elgamel and Bayoumi (2003), the crosstalk effect is only noticeable in the sub-micron technology (i.e., 0.25 \textmu m or below), and we simulate the wire propagation delay and the delay of bus codec with TSMC CMOS 0.18-\textmu m technology by HSPICE. In our simulations, the circuit model used for the interconnect wire is the \pi RC circuit model. The used circuit model for the gates is the MOSIS MOS circuit model. We suppose that all drivers and receivers have a uniform size, and all signal wires and routes have uniform length, width, and spacing. In the 0.18-\textmu m technology, the length, width, thickness, and spacing of the

![Figure 12. Comparisons of total power consumptions for different bus coding methods with the UMC 0.09-\textmu m CMOS technology for various widths: (a) 8-bit, (b) 16-bit, and (c) 32-bit.](image-url)
un-coded signal wire are 1300, 0.99, 0.53, and 1.37-μm, respectively. However, the length, width, thickness, and spacing of the coded signal wire are 1300, 0.72, 0.53, and 0.96-μm, respectively. We suppose that synchronous latches are located at the transmitter side. Thus, all signals switch at the same time on buses.

The case of a 32-bit bus is adopted as an example for analyses. Table 4 shows the propagation delay of wire and different bus codecs. The wire propagation delay is reduced by exploiting bus coding methods. The methods in Kim et al. (2000), Micea and Wayne (1995), Youngsoo et al. (1998, 2001b), Zhan et al. (2002), Khan et al. (2006) and Duan et al. (2001), and the proposed CDBI and ECDBI methods codec reduce wire propagation delays by 24.3%, 29.7%, 51.4%, 42.2%, 59%, 48.7%, 59.5%, and 62.2%, respectively. The total propagation delay is the summary of the wire propagation delay without the worst crosstalk patterns, the encoder delay, and the decoder delay. The total propagation delay of using the proposed methods is smaller than that of the other coding methods. The reduction ratio of delays is evaluated by comparing total propagation delay among different bus coding methods with the un-coded value. The proposed CDBI coding method reduces total propagation delays by 31.8% and 34.2%, respectively. The proposed ECDBI coding method reduces total propagation delay more than previous bus coding schemes do. The proposed CDBI coding method has the same coding efficiency as the Khan et al. (2006) method but the chip area of the CDBI method is smaller than that of Khan’s method. The experimental results show that the CDBI coding method reduces coupling activity by from 25.7% to 36.4% and switching activity by from 4.5% to 8.5% on 8-bit to 32-bit data buses, respectively. It reduces total power dissipation more than the other bus coding methods when a load capacitance is more than 0.3 pF/bit with UMC 0.09-μm CMOS technology. Furthermore, the proposed ECDBI coding method reduces coupling activity by from 28.4% to 38.4% and switching activity by from 10.1% to 14% on 8-bit to 32-bit data buses, respectively. It reduces total power dissipation more than the other bus coding methods when a load capacitance is more than 0.2 pF/bit with UMC 0.09-μm CMOS technology. For a 0.8 pF/bit load capacitance, both the proposed methods reduce total power use by 19.3–30.9% when systems are implemented with UMC 0.09-μm CMOS technology. Similarly, both the proposed methods also reduce total power dissipation more than the other bus coding methods with TSMC 0.18-μm CMOS technology. Meanwhile, the CDBI and the ECDBI schemes reduce total propagation delay by up to 31.8% and 34.2%, respectively, on 32-bit data buses.

5. Conclusion

Two novel bus-coding methods are proposed to reduce dynamic power dissipation and wire propagation delay on buses efficiently. The proposed bus coding methods reduce dynamic power dissipation and wire propagation delay more than previous bus coding schemes do. The proposed CDBI coding method has the same coding efficiency as the Khan et al. (2006) method but the chip area of the CDBI method is smaller than that of Khan’s method. The experimental results show that the CDBI coding method reduces coupling activity by from 25.7% to 36.4% and switching activity by from 4.5% to 8.5% on 8-bit to 32-bit data buses, respectively. It reduces total power dissipation more than the other bus coding methods when a load capacitance is more than 0.3 pF/bit with UMC 0.09-μm CMOS technology. Furthermore, the proposed ECDBI coding method reduces coupling activity by from 28.4% to 38.4% and switching activity by from 10.1% to 14% on 8-bit to 32-bit data buses, respectively. It reduces total power dissipation more than the other bus coding methods when a load capacitance is more than 0.2 pF/bit with UMC 0.09-μm CMOS technology. For a 0.8 pF/bit load capacitance, both the proposed methods reduce total power use by 19.3–30.9% when systems are implemented with UMC 0.09-μm CMOS technology. Similarly, both the proposed methods also reduce total power dissipation more than the other bus coding methods with TSMC 0.18-μm CMOS technology. Meanwhile, the CDBI and the ECDBI schemes reduce total propagation delay by up to 31.8% and 34.2%, respectively, on 32-bit data buses.

Acknowledgments

This study was supported by the National Science Council, Taiwan, R.O.C., under grant NSC96-2220-E-005-006. The authors thank the anonymous reviewers whose careful reviews and detailed comments have helped to improve the readability of this article and the National Chip Implementation Center (CIC) in Taiwan for providing the related EDA tools.
### Nomenclature

- $b(t)$: the original data
- $B(t)$: the complement value of $b(t)$
- $B(t-1)$: the encoded bus value to be sent on bus lines at time $t-1$
- $B(t)$: the encoded bus value to be sent on bus lines at time $t$
- $C_C$: the coupling capacitance
- $C_L$: the load capacitance
- $C_{eff}$: the coupling capacitance variation.
- $E$: the power supply voltage, equaling the rail-to-rail signal voltage in CMOS circuits
- $f$: clock frequency
- $INV$: an extra control bit
- $P_{D,\text{un-coded}}$: the dynamic power consumption on a bus with uncoded data
- $P_{D,\text{coded}}$: the dynamic power consumption on a bus with encoded data
- $P_{\text{enc}}$: the power consumption at the bus encoder
- $P_{\text{dec}}$: the power consumption at the bus decoder
- $T_S$: the average values of switching activity for load capacitance
- $T_C$: the average values of coupling activity for coupling capacitance
- $V_{DD}$: the supplying voltage
- $\lambda$: the capacitance ratio
- $\Delta V_1$: the voltage variations of the adjacent wires
- $\Delta V_2$: the voltage variation of the center wire
- $\Delta V_3$: the voltage variations of the adjacent wires

### References


busses. *IEEE transactions on very large scale integration systems*, 7 (2), 212–221.


