國立交通大學

資訊科學與工程研究所

碩士論文

考慮耦合電容以降低功率消耗的資料傳輸編解碼之有效方法

An Effective Amount-Driven Encoding/Decoding Method (ADEM) for Low-Power Data Bus with Coupling

研 究 生：蔡明憲

指導教授：陳　正　教授

中華民國九十五年六月

考慮耦合電容以降低功率消耗的資料傳輸編解碼之有效方法

An Effective Amount-Driven Encoding/Decoding Method (ADEM) for Low-Power Data Bus with Coupling

研究生：蔡明憲　　Student：Da-Ming Chang

指導教授：陳正 教授　　Advisor：Shyan-Ming Yuan

國立交通大學

資訊科學與工程研究所

碩士論文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

考慮耦合電容以降低功率消耗的資料傳輸編解碼之有效方法

研究生：蔡明憲　　　　指導教授：陳　正　教授

國立交通大學資訊科學與工程研究所碩士班

摘要

由於製程的進步，電路愈益精密，匯流排的長度也越來越長。而在匯流排中，線路間的距離也越來越短，其間所產生的寄生電容值則增大，使得其每次的充放電事件消耗了更多的電能。因此，如何同時降低自生電容與耦合電容在匯流排中所消耗的電能，成為一個很重要的課題。我們在本篇論文中，提出一個方法 *amount-driven encoding method* (ADEM)，針對 on-chip 的資料匯流排所消耗的電能進行改善，藉由整合資料編碼與 Spacing 的技術來達到此目標。我們將匯流排中的線路視為兩兩成對的相鄰線路，且沒有交集。再利用 Spacing 的技術以減小此成對的線路間所產生的耦合電容值，並對此成對之線路中的資料以四種編碼方式進行編碼，同時降低在此成對線路中自生電容與耦合電容所發生的充放電事件次數，進而達到低功率消耗的目標。我們模擬常見的多媒體檔案在匯流排中傳輸所消耗的電能，結果顯示 ADEM 在低功率消耗的效能上達到了 25%的改進。相較於過去的相關研究，除了節省了更多的耗能，我們在電路複雜度與時間延遲方面，也有較好的表現。

An Effective Amount-Driven Encoding/Decoding Method (ADEM) for Low-Power Data Bus with Coupling

Student: Ming-Hsien Tsai Advisor: Prof. Cheng Chen

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract

As technology trends advancing, the increased bus length and the narrower geometrical proximity of adjacent bus lines form non-negligible coupling capacitances between two adjacent bus lines. Therefore, more power dissipation is caused by charge and discharge of the coupling capacitances. In this case, the effect of line-to-ground and coupling capacitances plays an important role for low-power bus system. In this thesis, we propose an integrated method, named *amount-driven encoding method* (ADEM), which minimizes the power dissipation of on-chip data buses through combining bus encoding and Spacing mechanisms. In our bus model, the bus lines are considered as the constitution of several adjacent pairs without intersection. Spacing mechanism is applied to decrease the values of coupling capacitances between pairs. For coupling capacitances between two adjacent lines within a pair, we reduce the charge and discharge times of them by applying four encoding methods in each bus cycle. Our method saves more than 25% of bus power on average compared to the un-encoded cases by transferring a large set of common used multimedia files on the bus. Comparing to previous work, ADEM saves more power effectively with a little overhead of circuit complexity and delay time.

Acknowledgements

I wish to thank my advisor, Professor Cheng Chen, whose inspiration led to the development of this text. Without his guidance and encouragement, I could not finish this thesis. I also thank Professor Chih-Chun Shann and Dr. Kuan-Chou Lai for their comments.

My thanks to Che-Ying Liao who is delightful fellows, I felt happy and relaxed because of his presence. Especial thanks are due to Yi-Hsuan Lee, who has devoted so much time to reading and checking this thesis.

Finally, I am grateful to my dearest family for their encouragement. I also send my sincere thanks to my best friends. These include Su-Chen Chiu, Hung-Wei Wang, Shih-Yi Liao, Pang-Han Kao, and Fu-Tung Kao. They accompany me all the time.

Table of Contents

摘要 i

Abstract.......... ii

Acknowledgements iii

Table of Contents.......... iv

List of Figures.......... v

List of Tables vi

Chapter 1 Introduction 1

Chapter 2 Fundamental Background and Related Work.......... 4

2.1. Fundamental background.......... 4

2.1.1. Bus model.......... 4

2.1.2 Power dissipation model 5

2.2. Related work 11

2.2.1. Address bus.......... 12

2.2.2. Instruction bus 12

2.2.3. Data bus.......... 13

2.2.3.1. Bus-Invert 14

2.2.3.2. Odd/Even Bus-Invert (OEBI) 14

2.2.3.3. Coupling-Based Bus-Invert (CBBI) 15

2.2.3.4. Fibonacci Coding.......... 15

2.2.4. Spacing, Shielding, and Swapping 16

Chapter 3 Amount-Driven Encoding Method (ADEM) 18

3.1. Overview.......... 18

3.2. Principle of ADEM 21

3.3. Overhead reduction 27

3.4. Spacing mechanism.......... 29

Chapter 4 Experimental Results.......... 32

4.1. Overview of simulation.......... 32

4.2. Results analysis 33

4.2.1. Power dissipation caused by informed lines 34

4.2.2. The impact of capacitance ratio, distance ratio, and bus width.......... 34

4.3. Overhead comparison 40

Chapter 5 Conclusions and Future Work 42

5.1. Conclusions.......... 42

5.2. Future work 43

Bibliography 45

List of Figures

Figure 2.1. The general and simplified two-line bus model 5
Figure 2.2. The capacitance ratio for different technology generations.odel 6
Figure 2.3. Charging/discharging events of self-capacitance. 7
Figure 2.4. Charging/discharging, and toggling events of coupling capacitance 8
Figure 2.5. An example to calculate power dissipation caused by self-, coupling transitions, and toggling events 10
Figure 3.1. Power dissipation for original word and codeword encoded OEBI and CBBI...... 19
Figure 3.2. A common low-power bus model with Spacing mechanism 19
Figure 3.3. Average power dissipation for OEBI and ADEM 22
Figure 3.4. An encoding example 25
Figure 3.5. Spacing architecture 30
Figure 4.1. The average power saving of ADEM, ADEM_2L, and ADEM_4L (λ=3.9, α=0, M=32) 35
Figure 4.2. The average power saving of ADEM, ADEM_2L, and ADEM_4L without considering informed lines (λ=3.9, α=0, M=32) 35
Figure 4.3. The average power saving under various capacitance ratios (α=0, M=32) 36
Figure 4.4. The impact of Spacing mechanism (λ=3.9, M=32) 37
Figure 4.5. The impact of bus width with and without considering power dissipation caused by informed lines (λ=3.9, α=0) 39

List of Tables

Table 2.1. Power analysis for self-transitions 7
Table 2.2. Power analysis for coupling transitions 9
Table 3.1. All the cases of power dissipation caused by pair transitions for OEBI 21
Table 3.2. The order of power dissipations caused by pair transitions 23
Table 3.3. All the cases of power dissipation caused by pair transitions with ordering, and the corresponding encoding states 24
Table 3.4. The four-state encoding table of ADEM 24
Table 3.5. The four-state decoding table of ADEM 26
Table 4.1. Parameters in our experiment 33
Table 4.2. The comparison of circuit area, delay time, and the number of informed lines 41

Chapter 1 Introduction

Systems-on-a-chip (SOC) is expected to reach capacities that exceed the one billion transistor milestone within the next couple of years [12]. As a result, we will face new problems in the design of such circuits. The complexity and the physical length of bus systems will lead to an increased power dissipation of an SOC [8-14, 16-20] More importantly, the closer geometrical proximity of adjacent bus lines will lead to the effects that are more relevant in technologies as advanced as 100 *nm* and beyond [12-13]. This is because two or more adjacent bus lines will form a coupling capacitance between them. This effect not only leads to crosstalk and delay effects, but also introduces power dissipation caused by coupling transition, i.e. the coupling capacitance is charged and discharged when there is a voltage swing between two or more bus lines [8-14, 16-20]. This effect takes place in addition to the line-to-ground capacitance of a bus line i.e., the capacitance between the bus line and substrate/ground. Hence, more power is dissipated by self-transition when the line-to-ground capacitance is charged and discharged [2-7, 15].

There are several means to diminish or at least reduce the effect caused by self- and coupling transitions. The first one is to widen the distance between bus lines (i.e. Spacing mechanism), so that the value of coupling capacitance can be decreased. However, it will cause the total area of the bus system grows [14, 17-18]. Next, place & route (P&R) tools can be used to avoid side by side routing of bus lines [12, 17-18]. Nevertheless, because a one billion transistor SOC with multiple bus system and long buses with many cores connected to them is complex, the complexity of routing problem will prevent a satisfying solution at a feasible routing time. Then, the geometrical shape of bus lines can be reshaped to reduce the effect of coupling transition [12]. For example, the cross-section shape can be made narrower such that the distance between bus lines increases without sacrificing the space for whole bus system. This approach effectively reduces the effect of coupling transition, but more power dissipation caused by self-transitions because the distance between bus lines and other metal layers is decreased. In addition to decrease the value of capacitance, we also can reduce the number of self- and coupling transitions to achieve low power dissipation. Bus encoding technique can be used to reduce the number of self- and coupling transitions by encoding data stream. Because it can be combined with other techniques listed above, we will focus on bus encoding method design. In this thesis, we propose a integrated method named *amount-driven encoding method* (ADEM) to reach the goal of power reduction for on-chip data buses.

For designing bus encoding techniques, it's difficult to take all line-to-ground and coupling capacitances into consideration at the same time [8-10]. The reason is that the bit transition of a bus line accompanies effect upon two near by coupling capacitances. If we focus on one coupling transition, there may cause some influence on the adjacent one. Therefore, in this thesis, we consider the bus lines as the constitution of several adjacent pairs without intersection. In our bus model, we will apply spacing mechanism to decrease the values of coupling capacitances between pairs. For coupling capacitances between two adjacent lines within a pair, we will design an encoding method to reduce the number of coupling transitions. In our encoding method, different to previous work [2-3, 9, 13], ADEM first recognizes the type of bit-streams transferred on each pair. Then, it concurrently applies four encoding methods according to the appearance number of each type in one bus cycle. By using the integrated method composed of bus encoding and Spacing mechanisms, we can reduce the power dissipation of bus system effectively.

For evaluating the performance, we compare our method with previous work under various fabrications, distances between two pair of lines, and bus widths. The benchmarks used in our experiment are the multimedia file because they are common used in handheld device. From experiment result, we can observe that while the fabrication trends advancing, our method will save more power. For high-performance devices which contain wider bus, our ADEM is more suited than previous work [1, 9, 13]. Moreover, the complexity of encoding circuit is simple because our method doesn’t need to calculate total power dissipation during encoding.

This thesis is organized as follows. Chapter 2 introduces the bus model and power model, and then reviews some related work. Chapter 3 describes our ADEM in some detail. Performance evaluations are presented in Chapter 4. Finally, some conclusions and future work are given in Chapter 5.

Chapter 2 Fundamental Background and Related Work

In this chapter, we will present the bus model used in our experiments, and the power model for calculating power dissipated by data transferred on the bus. Furthermore, we will briefly survey the related works about bus encoding techniques and some non-encoding mechanisms for lowering power dissipation.

2.1. Fundamental background

2.1.1. Bus model

The bus model used in our scheme is proposed by [1] with some changes. This general two-line bus can be modeled as shown in Figure 2.1(a), where R_i is the internal resistance of the bus driver, r_l is the linear resistance of the bus lines, V_{dd} is the voltage of the power supply, c_l is the linear capacitance to the substrate (ground), c_c is the linear interwire capacitance between two adjacent lines, and c_{Load} is the capacitance introduced by the connection between bus lines and other devices. Nearly in all real bus models, the wire resistance is significantly smaller than the internal resistance of the bus driver. So that we can write: $r_l l_b << R_i$, where $l_b = \sum\Delta l$ is the length of a bus line. For the convenience of our experiment, we can sum up all the c_l and c_{Load} to a capacitance C_L and all the c_c to another capacitance C_C without lose of general. A simplified two-line bus model is illustrated with Figure 2.1(b), where $C_C = c_c l_b$ is the coupling capacitance (also known as inter-wire capacitance), and $C_L = c_L l_b + C_{\text{Load}}$ is the line-to-ground capacitance (also known as self capacitance or instinct capacitance). Although the bus model we introduced is a two-line one, it can be extended easily to a general n-line one.

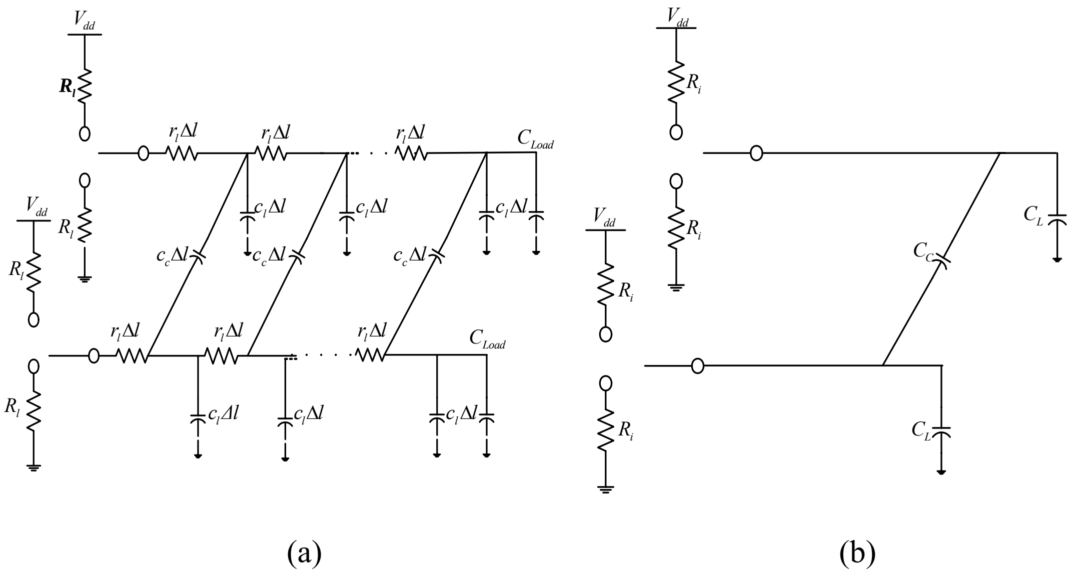


Figure 2.1. A two-line bus model. (a) Model of general two-line bus. (b) Simplified two-line bus model.

In the mean time, we assume that all the C_L on each line have the same value and so as C_C between the adjacent lines.

The relation between the coupling capacitance and line-to-ground capacitance is interesting. In some previous researches [2-6], the coupling capacitances C_C were disregarded and only the line-to-ground capacitances C_L were taken into account. However, while the fabrication shrinks, the capacitance ratio λ (λ=coupling capacitance/line-to-ground capacitance) grows, so that the coupling capacitance can no longer be disregarded. As illustrated from Figure 2.2 [8] the coupling capacitance is larger than the line-to-ground capacitance for modern fabrications.

2.1.2 Power dissipation model [9-12]

In the following, we will introduce how we calculate power dissipation used in our experiments. The transitions on the capacitances represent the change of the energy stored in the capacitances, i.e. occurrences of charging and discharging events. Furthermore, self- and coupling transitions are defined as transitions on the

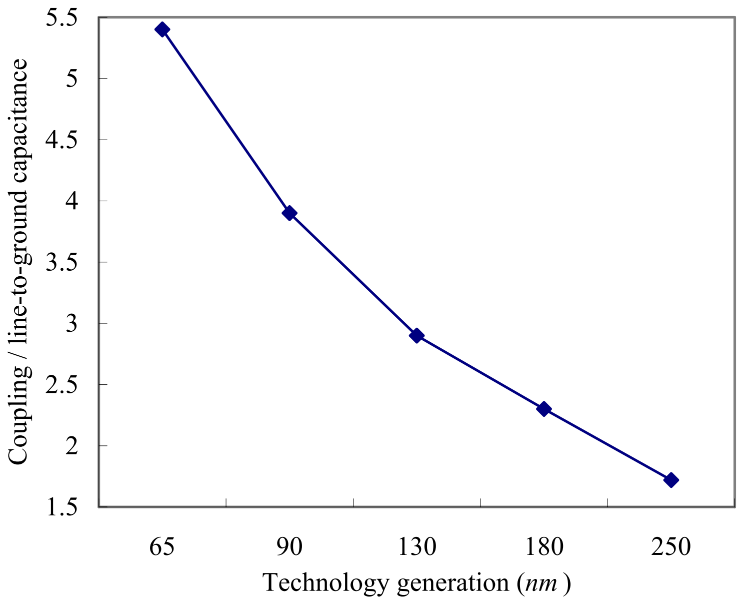
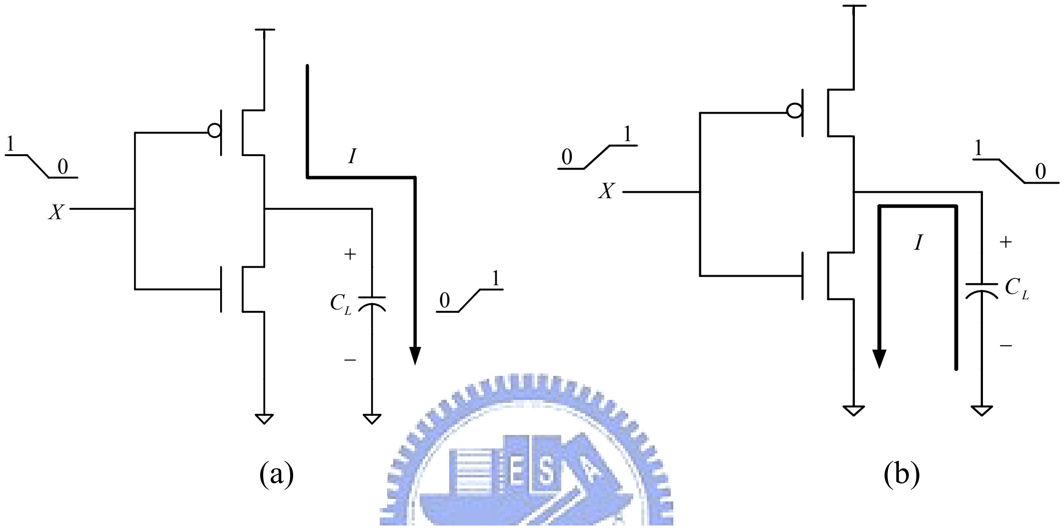


Figure 2.2. The capacitance ratio for different technology generations.

line-to-ground and coupling capacitance, respectively. These capacitances are charged/discharged during the transitions, which may introduce power dissipation. There has been some confusion in the literature about the difference between power consumption and power dissipation on buses. For power consumption, only the charging transitions are considered, and which require current flow from the power supply. Next, for power dissipation, all transitions need to be considered. In other words, if we calculate the energy consumed from the power supply, it introduces power consumption. If we calculate the energy dissipated caused by the transitions on the capacitances, it introduces power dissipation. In general, only the power consumption or the power dissipation needs to be calculated. This is because half of energy consumed from the power supply is stored in the capacitances, and which will be dissipated in the long run. Thus, the calculation of power consumption or power dissipation is equal on average, even if their instantaneous values are different. In this thesis, we focus on the transitions of the capacitances, so the power dissipation is adopted in order to give an exhaustive expression.

Sequence of bits	Events	Initial stored energy	Final stored energy	Energy dissipation	α_L
0→0	-	0	0	0	0
1→0	Charge	0	$C_LV^2/2$	$C_LV^2/2$	1
0→1	Discharge	$C_LV^2/2$	0	$C_LV^2/2$	1
1→1	-	$C_LV^2/2$	$C_LV^2/2$	0	0

Table 2.1. Power analysis for self-transitions.



(a)

(b)

Figure 2.3. (a)Charging (b)Discharging event of line-to-ground capacitance.

During the calculation, we have to make the following assumptions: 1) any capacitance will not be charged or discharged between two consecutive bus cycles and 2) the signals on the all lines have been synchronized, i.e. there is no delay between them. In the following, the dynamic energy dissipation per bus cycle of a bus line due to self-transition can be written as:

$$P_{ds} = \frac{1}{2} \cdot \alpha_L \cdot C_L \cdot V_{dd}^2 \qquad (1)$$

, where C_L and V_{dd} defined above is shown in the Figure 2.1(b) and α_L is the energy coefficient of self-transition of a bus line. Figure 2.3 shows the transitions, charging and discharging, on the line-to-ground capacitance. α_L is set to 1 when there exists a transition on the line-to-ground capacitance and 0 for else. Table 2.1 shows the energy stored in the line-to-ground capacitances before and after the transitions. In this table,

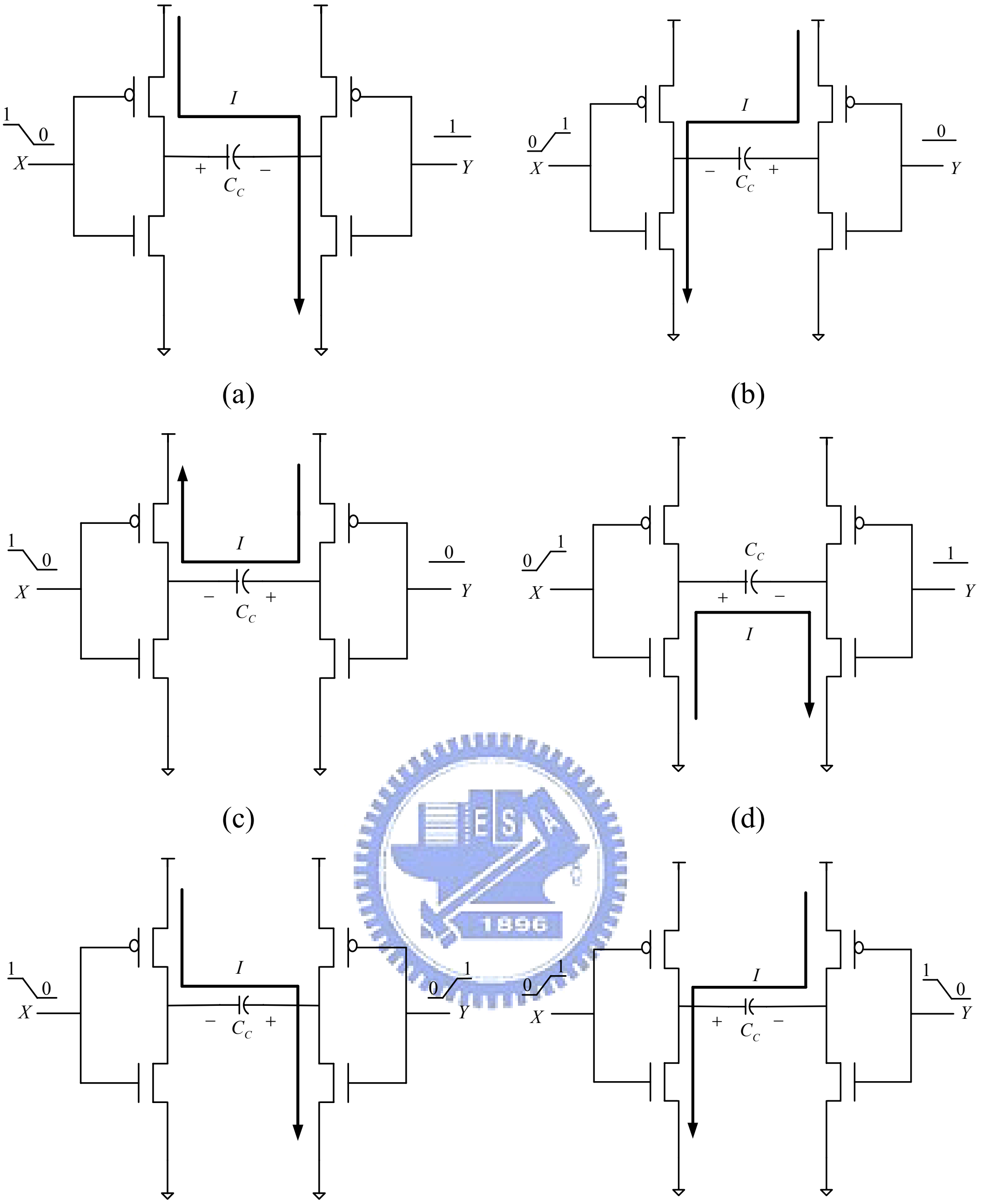


Figure 2.4. Charging/discharging, and toggling events of coupling capacitance. (a, b) Charge. (c, d) Discharge. (e, f) Toggle

‘0’ and ‘1’ represent the low and high electric potential of voltage, respectively. Then, 0→1 means a rising switching activity, and 1→0 means the falling one. The total power dissipation per bus cycle caused by self-transitions can be calculated by summing P_{ds} up for each bus lines.

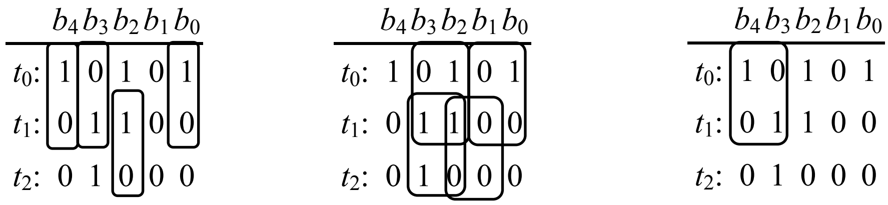
Sequence of bits	Events	Initial stored energy	Final stored energy	Energy dissipation	α_L
00→00	-	0	0	0	0
00→01	Charge	0	$C_CV^2/2$	$C_CV^2/2$	1
00→10	Charge	0	$C_CV^2/2$	$C_CV^2/2$	1
00→11	-	0	0	0	0
01→00	Discharge	$C_CV^2/2$	0	$C_CV^2/2$	1
01→01	-	0	0	0	0
01→10	Toggle	$C_CV^2/2$	$C_CV^2/2$	$2C_CV^2$	4
01→11	Discharge	$C_CV^2/2$	0	$C_CV^2/2$	1
10→00	Discharge	$C_CV^2/2$	0	$C_CV^2/2$	1
10→01	Toggle	$C_CV^2/2$	$C_CV^2/2$	$2C_CV^2$	4
10→10	-	0	0	0	0
10→11	Discharge	$C_CV^2/2$	0	$C_CV^2/2$	1
11→00	-	0	0	0	0
11→01	Charge	0	$C_CV^2/2$	$C_CV^2/2$	1
11→10	Charge	0	$C_CV^2/2$	$C_CV^2/2$	1
11→11	-	0	0	0	0

Table 2.2. Power analysis for coupling transitions.

The dynamic power dissipation per bus cycle between the neighboring bus lines due to coupling transitions can be written as:

$$P_{dc} = \frac{1}{2}\cdot \alpha_C \cdot C_C \cdot V_{dd}^2 \qquad (2)$$

, where α_C is the energy coefficient of coupling transitions. The charging and discharging of coupling capacitances display more cases. Figure 2.4 shows the possible cases of charging, discharging, and toggling of the coupling capacitances, and Table 2.2 shows the energy stored in the coupling capacitances before and after the transitions, where X_1Y_1→X_2Y_2 means a line X and the adjacent line Y which exhibit the X_1→X_2 and Y_1→Y_2 switching activities, respectively. In the cases of charging, energy of $C_CV_{dd}^2$ is supplied by the source. Half of this energy is dissipated in the circuit while the rest is stored in the coupling capacitance. In the cases of discharging, no energy is supplied by



(a) (b) (c)

Figure 2.5. An example to calculate power dissipation caused self-, coupling transitions, and toggling events. (a) 4 self-transitions. (b) 4 coupling transitions (without toggling). (c) 1 toggling events

the source. As the capacitance discharges, its stored energy of $1/2C_CV_{dd}^2$ is dissipated in the circuit. Furthermore, toggling is defined as the case where adjacent lines switch simultaneously in opposite directions. In this case, the relative change in the potential difference of the capacitance is $2V_{dd}$, thus the energy supplied by the source during the transition will be twice than the charging cases, i.e. $2C_CV_{dd}^2$. The final energy stored in the capacitance is the same as the initial value, and thus the total power dissipated in the circuit will be the same as that supplied by the source. The dissipated energy, $2C_CV_{dd}^2$, is four times than those of charging and discharging. The values of energy coefficient α_C corresponding to all possible 16 switching cases are listed in the table 2.2. The total power dissipation per bus cycle caused by coupling transitions can be calculated by summing P_{dc} up between each bus lines. Finally, total power dissipation per bus cycle caused by self- and coupling transitions P_d can be calculated by the following formula: $P_d=\sum P_{ds}+\sum P_{dc}$.

In the following, an example of how to calculate power dissipation is illustrated. Figure 2.5 presents the power dissipation caused by self- and coupling transitions on the 4-bit bus lines, where t_i is the time slice of bus cycles, and b_i is the fixed (physical) order of the bit-lines of the bus (the orders from left to right are [b_3, b_2, b_1, b_0]). Furthermore, the amount of self- and coupling transitions will cause the power dissipation, which is according to the α_L and α_C. In Figure 2.5(a), the number of bit-transitions will introduce six self-transitions activities, and thus the power dissipated by self-transitions is $\mathbf{6}\cdot 1/2C_LV_{dd}^2$. In Figure 2.5(b), the number of adjacent pairs of bit-transitions will introduce four coupling-transitions activities without the cases of toggling. Finally, in Figure 2.5(c) there exist two toggling events. Thus, the dissipated power by coupling-transition is $\mathbf{4}\cdot 1/2C_CV_{dd}^2 + \mathbf{2}\cdot 4\cdot 1/2C_CV_{dd}^2$. Consequently, when we set the capacitance ratio λ to 4, the total amount of dissipated power during bus cycle t_0 and t_3 is:

$$P_d = P_{ds} + P_{dc} = \frac{1}{2}\left(\sum \alpha_L C_L + \sum \alpha_C C_L \lambda\right)V_{dd}^2 = 21 \cdot C_L V_{dd}^2 \qquad (3)$$

As listed in formula (3), although toggling events only happen twice, they have consumed almost 60% of total dissipated power. Actually, power dissipation caused by toggling events is four times than those of others on coupling capacitance, and more than four times than that of self-transitions even. In the above example, it takes the largest proportion in total dissipated power. What we focus on is to decrease the number of self- and coupling transitions simultaneously in order to reduce total power dissipation caused by data transferred on the bus. There are some related works for designing a low-power bus scheme described in the following section.

2.2. Related work

In the past decades, many bus encoding techniques have been proposed to establish a low-power consumed bus model. With different types of data streams transferred on the buses, we can classify buses to three kinds: address bus, instruction bus, and data bus. Features of these buses are intrinsically different. In this section, we'll state the bus features and related works for them. Then, some bus encoding mechanisms focusing on data bus will be deeply discussed in the following. Besides, a few non-encoding mechanisms contained Spacing, Shielding, and Swapping will be mentioned.

2.2.1. Address bus

Data streams transferred on address bus are memory addresses. So, data streams on the bus provides with the characteristic of sequential access to memory and the locality of memory reference. A multitude of techniques for low-power address bus encoding take these characteristics into account. Benini *et al.* [5] proposed a prediction scheme T0 which used the above characteristics. An additional line is used to inform the memory controller. While the additional line is asserted, the memory controller computes the new memory address by simply incrementing the previous one. Aghaghiri *et al.* [6] further eliminated the requirement for an additional redundant control line. This line is replaced by sending out the same address with previous one to the memory controller. Then, any new address transmission is sufficient in recognizing such that the address incrementing mode is no longer in effect. The incrementing mode is replaced by transmitting the same address. Musoll *et al.* [15] proposed the working-zone encoding (WZE) scheme which took the locality of memory references into account. The basic idea is that programs favor a few working zones of their address space. So, if the encoder and the decoder both keep a base addresses table, the actual memory address can be expressed as an offset along with the base address in the table. The offset address is shorter than that of actual memory address, thus the transitions can be decreased. All above approaches effectively use the highly regular patterns in order to decrease the number of self- and coupling transition activities. For uncorrelated data streams, the applicability of these approaches would be highly limited.

2.2.2. Instruction bus

For instruction bus, data streams transferred on the bus are much irregular compared to address buses. However, instructions exhibit fixed format still, so that different format spaces can apply different encoding strategies. Also, instructions can be rescheduling in order to economize power dissipation. Benini *et al*. [4] proposed a methodology for low-power instruction set architecture (ISA) encoding. The adjacency of instructions is observed from simulating a set of applications. If two instructions frequently encounter in adjacency, their opcode parts are set from minimizing the number of self-transition activities. Yang [7] uses instruction rescheduling to eliminate self-transition activities. While the power dissipation is reduced, the schedule length may be lengthened. Petrov and Orailoglu [10] encode the same bit line of several successive instructions by applying specific transformation function. These transformation functions are stored with a table in the decoder, and the original instructions can be recovered after applying the corresponding transformation functions. This method incurs longer delay from looking up the table, and needs large area from storing the table. Furthermore, Wong and Tsui [11] proposed an encoding scheme for decreasing the memorized information while decoding the codewords. They encode the instructions stored in the same memory block with the same encoding strategies. This approach not only reduces the transition activities, but also reduces the required area needed in decoding.

2.2.3. Data bus

For data bus, data streams are even more irregular than instructions, and need a different approach. Because it doesn't depend on any assumption about data regularity, the above approaches are not suitable for them. Previous work which targets to the low-power data bus design almost regards the distribution of data streams as uniformly [2-3, 9, 11, 13]. Besides, for handheld devices, data files transferred on the bus are usually multimedia files, and the size of data is relatively larger than those of memory address or instructions. So, there exist more transition activities in data bus implied more power dissipation. Hence, we'll focus on data bus, and expect to reduce power dissipation by decreasing the amount of transition activities occurred. In the following, we'll deeply discuss some related works focused on data bus.

2.2.3.1. Bus-Invert

Stan and Burleson [2] proposed the Bus-Invert method to minimize the self-transition activities. The basic idea is to transfer an inverted word through the bus whenever it can reduce the Hamming distance between this word and its predecessor. An additional bus line is inserted to indicate whether the word is inverted or not. An in-depth theoretical analysis of Bus-Invert method has been presented by Lin [15]. For buses with uniformly distributed data, an expected value analysis takes benefits around 10% for 32-bit buses in line with reported experimental results. However, encoding to reduce the self-transition activities is enough to reduce power dissipation in previous bus models, but for deep-submicron buses the coupling transition activities are needed to be considered. The related work considered coupling transition activities is also presented in the next several subsections.

2.2.3.2. Odd/Even Bus-Invert (OEBI)

Zhang *et al.* [9] proposed the Odd/Even Bus-Invert (OEBI) method, which was an extension from Bus-Invert, and further considered coupling transition activities. The coding technique uses the simple observation that coupling capacitances are frequently charged and discharged by coupling transitions. In this bus model, half of lines have an odd number and the others have an even number (if bus lines are numbered "in-order"). The coding system encodes current word to four candidates, which are composed of original current word, current word with odd lines inverted, with even lines inverted, and with all lines inverted. Two additional lines are used to indicate the odd and even lines inverted. Finally, the comparator picks out the candidate with the least coupling transition activities to regards it as codeword transmitted on the bus. Nevertheless, the comparator consists of a large set of adders, which needs large measure of area and gate delays.

2.2.3.3. Coupling-Based Bus-Invert (CBBI)

Ghoneima and Ismail [13] observed that it is costly to calculate the number of transition activities between the previous word and current codewords. They proposed the Coupling-Based Bus-Invert (CBBI) method focused on the toggling events, and encoded them as follows: 01→10 to 01→01 and 10→01 to 10→10. In order to make the codewords recoverable, while a line exhibits the rising switching activity and quiet at ‘0’ state, we encode it to quiet at ‘0’ state and the rising switching activity, respectively (i.e. 0→1 to 0→0 and 0→0 to 0→1). Similarly, while it exhibits the falling switching activity and quiet at ‘1’ state, we encode it to quiet at ‘1’ state and the falling switching activity (i.e. 1→0 to 1→1 and 1→1 to 1→0).

Under the coding scheme, there exist some penalties for encoding 01→01 and 10→10 to 01→10 and 10→01, respectively. In contrast with penalties, there exist some rewards for encoding 01→10 and 10→01 to 01→01 and 10→10, respectively. Thus, a simple decision circuit can be supplied to the comparator for choosing the words needed to be encoded. The delay and area of the data-path has increased. However, the trade-off delay versus power dissipation is common to all power minimization techniques.

2.2.3.4. Fibonacci Coding

Lindkvist *et al.* [8] introduced a ternary bus state representation, which could be used to construct a Fibonacci coding scheme without memory, i.e. the encoder didn’t need to record the previous word to encoding the next word. Under the ternary bus state representation, data transferred on the bus can be encoded without toggling events occurred. They transform the binary pair 00 as well as 11 into ‘0’, 01 into ‘+’, and 10 into ‘-‘. For example, the binary vector 0110 corresponds to the ternary vector +0-. Moreover, the coding scheme has to fulfill the following two conditions: 1) ‘+’ is only allowed in even coordinates and ‘-’ is only allowed in odd coordinates. 2) Neither two ‘+’ nor two ‘-’ may be adjacent, and zeros are disregarded. By fulfilling the first condition, there are no toggling events occurred during the transmission. Besides, the number of ternary vectors of length n-1, which are fulfilled the above conditions is *Fibonacci*(n+2).

Afterward, a heuristic method is used to choose the subset of the Fibonacci codewords to make the number of codewords be a power of two. Finally, another heuristic algorithm is used to map the original words and the subset of the Fibonacci codewords. The Fibonacci codewords are 1-to-1 mapping to the original words, so that the encoder can encode the original word without memorizing previous word. Although the bus width is enlarged in order to satisfy the width of Fibonacci codewords, the power dissipation can be reduced because no toggling events occurred.

2.2.4. Spacing, Shielding, and Swapping

Other than encoding methods, there are some technologies used for lowing power dissipation, including Spacing, Shielding, and Swapping. Spacing [14, 17-18] is a technique, which widens the distance between adjacent bus lines. As the distance widened, although the number of coupling transitions is not changed, the smaller coupling capacitances imply less power dissipation. Shielding [14] is a technique similar to Spacing, which inserts power/ground metal shields between adjacent bus lines to avoid the undesirable increases in coupling capacitances. In the meantime, Shielding also reduces inductive effects because of the closer return path to ground for the current flowing through signal lines. However, widening the distance or inserting the shield wires between every pair of signal lines is costly in area, which leads to increase the cost of the production. Arunachalam *et al*. [14] presented a comprehensive analysis between Spacing and Shielding. Under their analysis, the unnecessary shielding may significantly increase the value of coupling capacitances such that more power is dissipated. Thus, Spacing is usable to decrease power dissipation compared to Shielding.

Swapping [17-18] is a technique statically reordering the wires such that bus lines with a similar behavior are laid in adjacency. So, the coupling transitions are decreased due to Swapping. However, the swapped bus model should accompany a set of mapping functions in order to recover the original shape of the bus wires. These mapping functions have to be transferred on the bus to inform the decoder, so that incur the cost in delay and power dissipation. The optimal swapping problem is NP-hard [17-18], so it's unsuitable to practice in run-time.

In the next chapter, we will describe a new cost-effective low-power bus model which combines encoding and Spacing techniques. In our bus model, both self- and coupling transitions are reduced so that the total power dissipation can be reduced more compared with previous.

Chapter 3 Amount-Driven Encoding Method (ADEM)

From the related work described above, we can observe that decreasing the number of self- and coupling transitions simultaneously is an important factor for reducing total power dissipation. In this chapter, we will focus our working on data bus, and expect to establish a new cost-effective low-power bus model. In section 3.1, we will introduce our motivations and give an overview of our proposed method. Then, the bus encoding method *amount-driven encoding method* (*ADEM*) is described in section 3.2. The strategy to reduce overheads caused by ADEM is shown in section 3.3. Finally, we will deeply discuss the Spacing mechanism in section 3.4, which further enhances ADEM.

3.1. Overview

There is a simple example to illustrate the flaws of OEBI and CBBI. Figure 3.1 presents the power dissipation before and after applying OEBI and CBBI. Without applying any encoding methods, the original data streams are presented in (a), which causes $\mathbf{28}\cdot 1/2C_LV_{dd}^2$ of power dissipation (assume λ=3). Figure 3.1(b) and (c) present the encoded data streams after applying OEBI and CBBI, respectively. The power dissipations are both decreased to $\mathbf{27}\cdot 1/2C_LV_{dd}^2$, but it's still not good enough. The reason is that the existent toggling event (marked with a frame) is not eliminated. Actually, for any bus encoding method, if we focus on improving one coupling transitions, there may cause some influences on the adjacent one. Furthermore, if we take all coupling transitions into consideration simultaneously, the improvement will be highly limited because it's difficult to optimize all of them only by a few additional

	b_4	b_3	b_2	b_1	b_0
t_0:	1	0	1	0	1
t_1:	0	1	1	0	0
t_2:	0	1	0	0	0

(a)

	b_0	b_1	b_2	b_3	b_4	*odd*	*even*
t_0:	1	0	1	0	1	0	0
t_1:	1	1	0	0	1	0	1
t_2:	1	1	1	0	1	0	1

(b)

	b_0	b_1	b_2	b_3	b_4	*Inv*
t_0:	1	0	1	0	1	0
t_1:	1	0	0	1	1	1
t_2:	1	0	1	1	1	1

(c)

Figure 3.1. Power dissipation. (a) Original, **28**·1/2$C_LV_{dd}^2$. (b) OEBI, **27**·1/2$C_LV_{dd}^2$. (c) CBBI, **27**·1/2$C_LV_{dd}^2$ (assume λ=3).

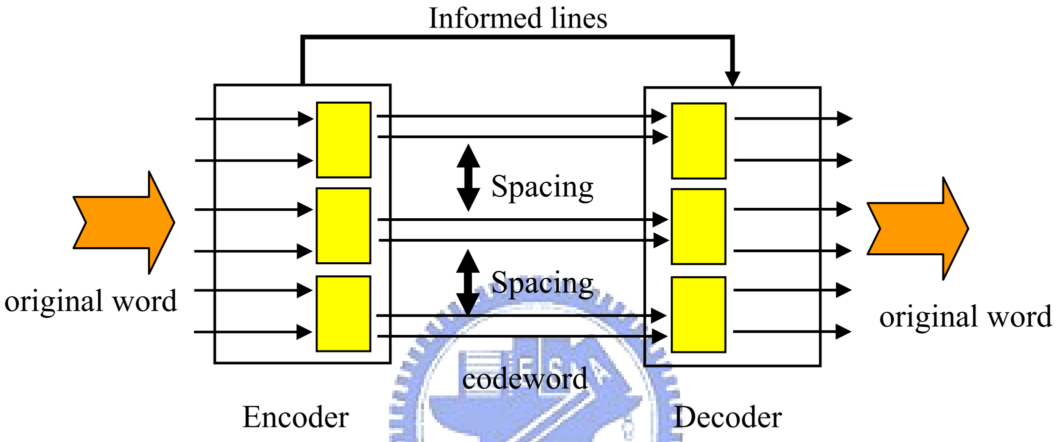


Figure 3.2. A common low-power bus model with Spacing mechanism.

lines. Thus, we will try to find an integrated method to deal with self- and coupling transitions at the same time. In our proposed method, we first use a new encoding method to handle them more effectively, and then apply Spacing mechanism to further resolve the difficulty of optimizing all coupling transitions simultaneously.

A common low-power bus model with a little difference is shown in Figure 3.2. The encoder receives the original data streams and then encodes them with a certain encoding method. In the meantime, it also has to inform the decoder what encoding criteria are used by using the informed lines. Then, the decoder recovers the codewords according to the information provided by the encoder. Meanwhile, Spacing technique is applied for each pair of lines. The detail of encoder, decoder, and Spacing technique will be introduced in the next three subsections. By using this integrated method, we can reduce power dissipation caused by the transmission of data streams effectively.

Before describing our bus encoding scheme in some detail, we introduce the following terminologies at first. The original input data streams at time t are represented by (b_0^t, b_1^t,…, $b_{M\text{-}1}^t$), and the encoded data streams also called codewords are represented by (b'^t_0, b'^t_1,…, $b'^t_{M\text{-}1}$), where M is the bus width. Then, we separate the data streams into several pairs before and after encoding. We name a pair among the original data streams at time t as $Pair_i^t$=(b_{2i}^t, b_{2i+1}^t) and an encoded pair among codewords as $EPair_i^t$=(b'^t_{2i}, b'^t_{2i+1}) for all $i\in\{0, 1,\ldots,M/2\text{-}1\}$. However, because there are two line-to-ground and one coupling capacitances for each pair of lines, we name a pair transition as $Pair_i^{t\text{-}1}\rightarrow Pair_i^t$ while there is self- or coupling transition on these capacitances. Thus, we denote the power dissipation caused by pair transition as $Power(Pair_i^{t\text{-}1}\rightarrow Pair_i^t)$. For instance, the power dissipated by pair transition $(0,1)\rightarrow(1,1)$ is calculated as $Power((0,1)\rightarrow(1,1))=1/2C_LV_{dd}^2+1/2C_CV_{dd}^2$. Besides, since there may occur only four types of pairs, i.e. (0,0), (0,1), (1,0), and (1,1), each encoded pairs will cause only four kinds of pair transitions. The average (expected) power dissipation for a certain pair is defined in the following formula:

$$Power_{avg.}(Pair_i^t) = Average\begin{pmatrix} Power((0,0)\rightarrow EPair_i^t), Power((0,1)\rightarrow EPair_i^t), \\ Power((1,0)\rightarrow EPair_i^t), Power((1,1)\rightarrow EPair_i^t) \end{pmatrix} \quad (4)$$

By using the formula, we can calculate total power dissipation and compare it to related work. Nevertheless, the coupling transitions between each pair of lines are still not taken into account, and these will be resolved by applying Spacing technique eventually.

In the following, we will introduce our new bus encoding method named ADEM in some detail. Meanwhile, to simplify the description we will omit the unit of power with $1/2C_LV_{dd}^2$.

$Pair_k^{t-1}\rightarrow Pair_k^t$	Events	α_L	α_C	OEBI			
				original	all invert	invert odd	invert even
(0,0)→(0,0)	-	0	0	0	2	**$1+\lambda$**	**$1+\lambda$**
(0,0)→(0,1)	Charge	1	1	**$1+\lambda$**	**$1+\lambda$**	0	2
(0,0)→(1,0)	Charge	1	1	**$1+\lambda$**	**$1+\lambda$**	2	0
(0,0)→(1,1)	-	2	0	2	0	**$1+\lambda$**	**$1+\lambda$**
(0,1)→(0,0)	Discharge	1	1	$1+\lambda$	$1+\lambda$	0	**$2+4\lambda$**
(0,1)→(0,1)	-	0	0	0	**$2+4\lambda$**	$1+\lambda$	$1+\lambda$
(0,1)→(1,0)	Toggle	2	4	**$2+4\lambda$**	0	$1+\lambda$	$1+\lambda$
(0,1)→(1,1)	Discharge	1	1	$1+\lambda$	$1+\lambda$	**$2+4\lambda$**	0
(1,0)→(0,0)	Discharge	1	1	$1+\lambda$	$1+\lambda$	**$2+4\lambda$**	0
(1,0)→(0,1)	Toggle	2	4	**$2+4\lambda$**	0	$1+\lambda$	$1+\lambda$
(1,0)→(1,0)	-	0	0	0	**$2+4\lambda$**	$1+\lambda$	$1+\lambda$
(1,0)→(1,1)	Discharge	1	1	$1+\lambda$	$1+\lambda$	0	**$2+4\lambda$**
(1,1)→(0,0)	-	2	0	2	0	**$1+\lambda$**	**$1+\lambda$**
(1,1)→(0,1)	Charge	1	1	**$1+\lambda$**	**$1+\lambda$**	2	0
(1,1)→(1,0)	Charge	1	1	**$1+\lambda$**	**$1+\lambda$**	0	2
(1,1)→(1,1)	-	0	0	0	2	**$1+\lambda$**	**$1+\lambda$**

Table 3.1. All the cases of power dissipation caused by pair transitions for OEBI.

3.2. Principle of ADEM

In the view of pair transition, the power dissipation caused by pair transitions for OEBI is listed in Table 3.1. There are four columns, including original, all invert, invert odd, and invert even, which are the four kinds of encoding methods in OEBI. The bold values stand for the critical cases, i.e. $Power(Pair_k^{t-1}\rightarrow Pair_k^t)$ is more than those with the same types of $Pair_i^{t-1}$. However, $Power(Pair_k^{t-1}\rightarrow Pair_k^t)$ with the same types of $Pair_k^t$ may be critical or non-critical case, so that $Power_{avg.}(Pair_k^t)$ is always equal to either $(4+2\lambda)/4$ or $(4+6\lambda)/4$ in OEBI. This balance distribution of average power dissipation for OEBI is shown in Figure 3.3(a). The reason for the balance is that OEBI always applies only one encoding method in each bus cycle. Nevertheless because the

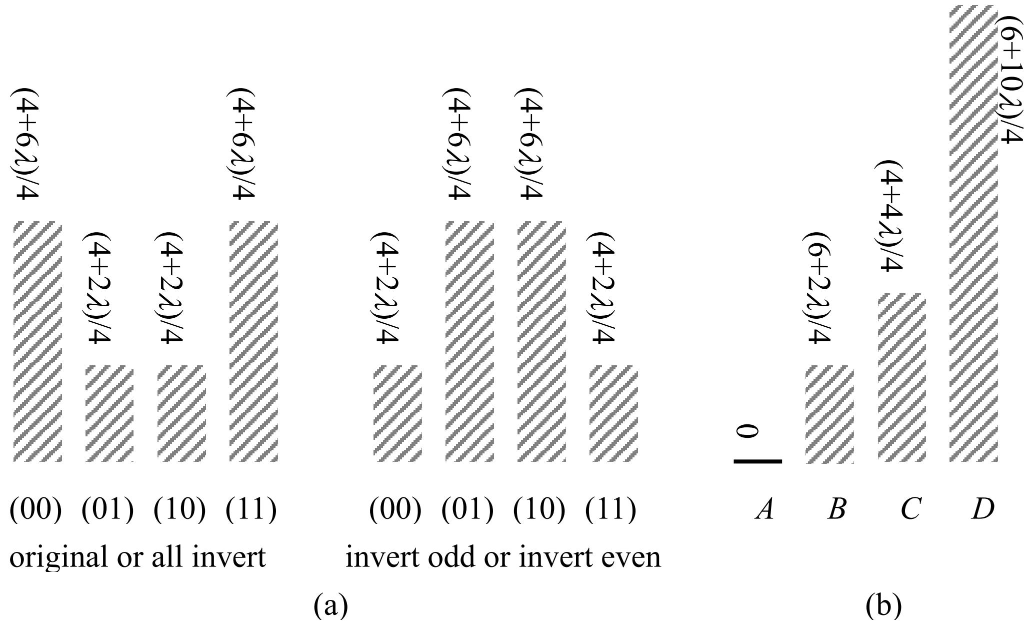


Figure 3.3. Average power dissipation for (a) OEBI and (b) ADEM.

number of pairs appeared in a bus cycle will not be balanced, this balance distribution of average power dissipation limits the improvement. Our modified unbalance distribution of average power dissipation is shown in Figure 3.3(b), where *A*, *B*, *C*, and *D* represent the types of pair appeared mostly, second mostly, third mostly and the rarely, respectively. In the following, we will show that the unbalance distribution of average power dissipation is more suitable than that of the balance one, and then explain the principle of our new encoding method more clearly.

Let the four types of pairs *A*, *B*, *C*, and *D* appear n_1, n_2, n_3, and n_4 times respectively, where $n_1 \geq n_2 \geq n_3 \geq n_4$. The appearance number of these types of pairs multiplied by those of average power dissipation will be the total power dissipation. Then, the total power dissipation of ADEM in a bus cycle can be estimated as:

$$P_d(\text{ADEM}) = n_1 \cdot 0 + n_2 \cdot \frac{6+2\lambda}{4} + n_3 \cdot \frac{4+4\lambda}{4} + n_4 \cdot \frac{6+10\lambda}{4} \qquad (5)$$

In OEBI, we assume that the pairs appeared mostly and second mostly have the smaller two average power dissipations, (4+2λ)/4. The pairs appeared rarely and second rarely have the larger two average power dissipations, (4+6λ)/4. Thus, the total power

$Pair_k^{t\text{-}1} \rightarrow Pair_k^{t}$	α_L	α_C	Power dissipation	Order
(0,0)→(0,0)	0	0	0	1
(0,0)→(0,1)	1	1	**1+λ**	3
(0,0)→(1,0)	1	1	**1+λ**	4
(0,0)→(1,1)	2	0	2	2
(0,1)→(0,0)	1	1	1+λ	3
(0,1)→(0,1)	0	0	0	1
(0,1)→(1,0)	2	4	**2+4λ**	4
(0,1)→(1,1)	1	1	1+λ	2
(1,0)→(0,0)	1	1	1+λ	2
(1,0)→(0,1)	2	4	**2+4λ**	4
(1,0)→(1,0)	0	0	0	1
(1,0)→(1,1)	1	1	1+λ	3
(1,1)→(0,0)	2	0	2	2
(1,1)→(0,1)	1	1	**1+λ**	3
(1,1)→(1,0)	1	1	**1+λ**	4
(1,1)→(1,1)	0	0	0	1

Table 3.2. The order of power dissipations caused by pair transitions.

dissipation of OEBI in a bus cycle can be expressed by the following formula:

$$P_d(\text{OEBI}) \geq (n_1 + n_2) \cdot \frac{4+2\lambda}{4} + (n_3 + n_4) \cdot \frac{4+6\lambda}{4} \qquad (6)$$

After subtracting formula (5) from (6), we can obtain that the total power dissipation of ADEM is always less than or equal to that of OEBI if we don’t consider the power dissipation caused by informed lines. In summary, the unbalance distribution of average power dissipation in ADEM is better than that of the balance one in OEBI. The design flow of this unbalance distribution is described as follows.

All the cases of $Pair_k^{t\text{-}1}$→$Pair_k^{t}$ are listed in the Table 3.2. The orders are listed according to $Power(Pair_k^{t\text{-}1}$→$Pair_k^{t})$ with the same type of $Pair_k^{t\text{-}1}$. We encode the pairs $Pair_k^{t}$ by arranging these orders in Table 3.3. The states are presented as the change between previous pairs $Pair_k^{t\text{-}1}$ and encoded pairs $EPair_k^{t}$. After arranging these pair

$Pair_k^{t-1}$→$Pair_k^t$	$EPair_k^t$	State	Power dissipation	Order
(0,0)→(0,0)	(0,0)	unchange	0	1
(0,0)→(0,1)	(1,1)	all invert	2	2
(0,0)→(1,0)	(0,1)	even invert	**$1+\lambda$**	3
(0,0)→(1,1)	(1,0)	odd invert	**$1+\lambda$**	4
(0,1)→(0,0)	(0,1)	unchange	0	1
(0,1)→(0,1)	(1,1)	even invert	$1+\lambda$	2
(0,1)→(1,0)	(0,0)	odd invert	$1+\lambda$	3
(0,1)→(1,1)	(1,0)	all invert	**$2+4\lambda$**	4
(1,0)→(0,0)	(1,0)	unchange	0	1
(1,0)→(0,1)	(0,0)	even invert	$1+\lambda$	2
(1,0)→(1,0)	(1,1)	odd invert	$1+\lambda$	3
(1,0)→(1,1)	(0,1)	all invert	**$2+4\lambda$**	4
(1,1)→(0,0)	(1,1)	unchange	0	1
(1,1)→(0,1)	(0,0)	all invert	2	2
(1,1)→(1,0)	(0,1)	even invert	**$1+\lambda$**	3
(1,1)→(1,1)	(1,0)	odd invert	**$1+\lambda$**	4

Table 3.3. All the cases of power dissipation caused by pair transitions with ordering, and the encoding states.

		A	*B*	*C*	*D*
Previous encoded pair	00	unchange	all invert	even invert	odd invert
	01	unchange	even invert	odd invert	all invert
	10	unchange	even invert	odd invert	all invert
	11	unchange	all invert	even invert	odd invert
		↓	↓	↓	↓
$Power_{avg.}(\cdot)$		0	$(6+2\lambda)/4$	$(4+4\lambda)/4$	$(6+10\lambda)/4$

Table 3.4. The four-state encoding table of ADEM.

transitions with those of orders, $Power_{avg.}(Pair_k^t)$ will be modified to 0, $(6+2\lambda)/4$, $(4+4\lambda)/4$, or $(6+10\lambda)/4$. Thus, unbalance distribution is obtained and the states listed in Table 3.3 can be employed to design our encoding method.

A four-state encoding table is listed in Table 3.4. The encoding states represent the

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
Previous codeword *t*-1:	0	0	1	0	1	1	0	0	1	1	0	1	1	0	1	1
Original word *t* :	0	1	1	0	0	0	1	1	0	1	1	0	0	1	0	0

⇩ After encoding (*A*=(0,1), *B*=(1,0), *C*=(0,0), *D*=(1,1))

	b'_0	b'_1	b'_2	b'_3	b'_4	b'_5	b'_6	b'_7	b'_8	b'_9	b'_{10}	b'_{11}	b'_{12}	b'_{13}	b'_{14}	b'_{15}
Previous codeword *t*-1:	0	0	1	0	1	1	0	0	1	1	0	1	1	0	1	1
Current codeword *t* :	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**

Figure 3.4. An encoding example (the energy dissipation caused by pair transitions is reduced from 12+10λ to 5+5λ).

change from previous encoded pair. This encoding table design follows the states in Table 3.3 and leads out the unbalance distribution of average power distribution. During the encoding, the encoder should record the previous codeword and encodes the input original word with following encoding steps:

I. Accounting the appearance number of each type of pair from the input original word.

II. Recognizing which types of pair appeared mostly, second mostly, third mostly, and rarely (i.e. which type of pair is *A*, *B*, *C*, and *D*) according to the accounted appearance number.

III. After recognizing the types of pair, the states can be found by referencing the four-state encoding table.

IV. Finally, the original pair with corresponding type *A*, *B*, *C*, and *D* can be encoded from previous encoded pair with the change of relative state.

After that, the codeword composed of encoded pairs will be obtained. Furthermore, the average power dissipation of these encoded pairs will be modified to 0, (6+2λ)/4, (4+4λ)/4, and (6+10λ)/4. Meanwhile, in order to make the codeword recoverable, the types of pair *A*, *B*, *C*, and *D* have to be transferred to decoder. Actually, the encoder only

		states			
		unchange	odd invert	even invert	all invert
Previous encoded pair	00	A	D	C	B
	01	A	C	B	D
	10	A	C	B	D
	11	A	D	C	B

Table 3.5. The four-state decoding table of ADEM.

transfers the orders of appearance number for each type of pair, which are totally 4!=24 cases.

An encoding example is shown in Figure 3.4. For the input original word, the types of pair (0,1), (1,0), (0,0), and (1,1) appear 3, 2, 2, and 1 times ,respectively. Thus, The encoder can recognize the types of pair as A=(0,1), B=(1,0), C=(0,0), and D=(1,1). By referencing the encoding table, the first pair $Pair_0^t$=A=(0,1) is encoded to $EPair_0^t$=(0,0) because the unchange state stands for leaving previous pair $Pair_0^{t-1}$=(0,0) unchanged. The remained pairs also can be encoded by repeating the encoding step III and IV. After all pairs in original word are encoded, the codeword (with bold values) is obtained and transferred on the bus. According to formula (3) (described in chapter 2) without calculating coupling transitions between each pair of lines, the power dissipation by pair transitions is reduced from 12+10λ to 5+5λ.

During the decoding, the decoder also should record the previous codeword and recovers the original words according to the following decoding steps:

I. The state can be obtained by observing the change between previous and current codewords. For instance, while the previous encoded pair is (1,0) and the current one is (0,0), the state can be gotten as even invert.

II. The decoder also should recognize what types are the types of A, B, C, and D, which are informed by encoder.

III. After finding the states and recognizing the types, decoder can recover the original pairs by referencing the decoding table listed in Table 3.5. The original word is obtained until all the encoded pairs are recovered.

A decoding example is the same as that of Figure 3.4. The first encoded pair $EPair_0^t$=(0,0) can be recovered back to A=(0,1) because the unchange state is recognized and previous encoded pair $EPair_0^{t-1}$=(0,0). The second one $EPair_2^t$=(0,0) is recovered back to B=(1,0) according to even invert state and $EPair_2^{t-1}$=(1,0). Else of them also can be recovered by repeating the decoding steps II and III.

We have introduced our encoding and decoding steps above. In order to recovery, there are a little informed lines should be inserted. Nevertheless, these informed lines still cause power dissipation, thus we will focus our working on reducing the number of them in the next section.

3.3. Overhead reduction

To realize our low-power encoding method ADEM, the encoder should inform the decoder the types of pairs A, B, C, and D. We need to insert five additional informed lines to record the 4!=24 cases of the types of pairs. However, these informed lines are costly not only in power dissipation but also in circuit area, and they will counteract our achieved performance. By observing formula (5), we find that there exist two critical elements, i.e. 0 and $(4+10\lambda)/4$, which are the smallest and largest values of average power dissipation for a pair. Furthermore, these two critical elements are the average power dissipation of the types of pair A and D. Here, we focus our working on the overhead of power dissipation caused by informed lines. In our first policy of overhead reduction named *ADEM_4L*, we will only recognize the types of pair A and D instead of recognizing all types of pairs. In the second one named *ADEM_2L*, we will only recognize the types of pair D because the element $(4+10\lambda)/4$ occupies intolerable large portion of total power dissipation. By using ADEM_4L and ADEM_2L, the additional informed lines can be reduced.

Under ADEM_4L, the encoder only recognizes the types of pair *A* and *D* and then informs the decoder. It is amounted to $\mathrm{P}_2^4 = 12$ cases so that only four additional informed lines will be needed. The encoding and decoding steps in ADEM_4L are almost the same as those in ADEM. The difference is that the encoding step II is modified as follows. The encoder only recognizes which types of pair appeared mostly and rarely (i.e. which types of pair are *A* and *D*) instead of recognizing all the types of pair. Next, the decoding step II is modified as follows. The decoder only recognizes what types are the types of *A* and *D* without recognizing all types of pair. Actually, the decoder does not obtain which types of pair are *B* and *C* from encoder. Therefore, we set the four types of pair to a specific sequence: (0,0), (0,1), (1,0), (1,1). While the types of pair *A* and *D* are recognized, they should be removed from the specific sequence. The types of pairs *B* and *C* can be simply set to the former and later of the rest two types according to the specific sequence. Meanwhile both encoder and decoder should follow the above rules to set the types of pair *B* and *C* for consistency. Since all the types of pair have been decided, both encoder and decoder can continue the following encoding and decoding steps.

For instance, the input original word and previous codeword are the same as those in Figure 3.3. The encoder only recognizes the types of pair appeared mostly *A* as (0,1) and rarely *D* as (1,1) at first. Meanwhile, the specific sequence of four types turns from (0,0), (0,1), (1,0), (1,1) into (0,0), (1,0). The types of pair *B* and *C* will be set to the former (0,0) and later (1,0), respectively. Finally, by referencing the same encoding and decoding tables like ADEM, the codeword can be obtained and recovered.

Under ADEM_2L, the encoder only recognizes the types of pairs *D*. There are only two additional informed lines will be inserted due to $\mathrm{P}_1^4 = 4$ cases. Similar to

ADEM_4L, the encoding and decoding steps in ADEM_2L are almost the same as those in ADEM. The difference is that both encoder and decoder only recognize the type of pair appeared rarely. Although the encoder and decoder do not get which the types of pair are *A*, *B*, and *C*, they can be picked out according to the specific sequence where the types of pair *D* has removed. In this case, the types of pair *A*, *B* and *C* can be set to the former, middle, and later types of the rest three types. In summary, since the encoder doesn't recognize all the types of pair in both ADEM_4L and ADEM_2L, the order of average power dissipation cannot match the number of the pairs appeared properly. Hence, the performance of ADEM_4L and ADEM_2L may not as good as ADEM.

We have reduced the 5 additional informed lines to 4 and 2 for recovery in ADEM_4L and ADEM_2L, respectively. Both the number of self- and coupling transitions caused by the informed lines can be decreased so that the overhead of power dissipation also can be reduced. Besides, because the encoder doesn't need to recognize all the types of pairs, the complexity of encoding circuit can be simplified. Also because of that, it is difficult to analyze the total power dissipation like ADEM. However, we will use simulations to get the results of performance and overhead for ADEM_2L and ADEM_4L. The overall evaluation will be given in chapter 4.

3.4. Spacing mechanism

The coupling transition between two pair of lines is presented as $b'^{t-1}_{2i+1}b'^{t-1}_{2i+2} \rightarrow b'^{t}_{2i+1}b'^{t}_{2i+2}$. If we apply any other encoding method to deal with it, the values of b'^{t}_{2i+1} and b'^{t}_{2i+2} may be changed and causes some influences on the encoded pairs. Thus, we have to use the non-encoding methods to deal with them. There are three non-encoding methods listed in our survey, i.e. Spacing, Shielding, and Swapping. However, it has been proven that Shielding can be replaced by Spacing if inductive effects are not

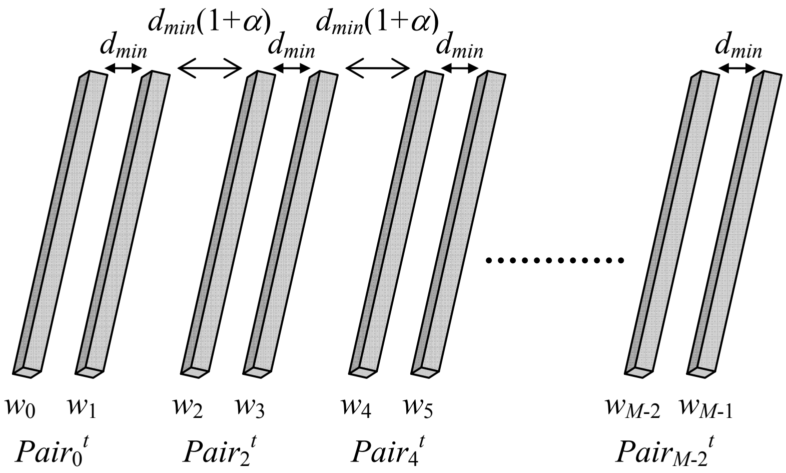


Figure 3.5. Spacing architecture.

considered [14]. Actually, power dissipation is our major concern here instead of inductive effects. Besides, as mentioned in Section 2.2.4 Swapping is not suitable to practice in run time. Therefore, Spacing seems to be the practicable choice to deal with the coupling transitions between two adjacent pairs.

As mentioned in our survey, the value of coupling capacitance depends on the distance between two adjacent wires. While the distance is widened, the value will be decreased. Furthermore, for a simple bus layout, the coupling capacitance between two neighboring wires can be estimated by [18]:

$$C_C = \varepsilon \cdot \frac{A}{d} = \varepsilon \cdot \frac{A}{d_{min}(1+\alpha)} \tag{7}$$

, where A is the contact area between two neighboring wires which is dependent on the height and length of the wires, d is the distance between the wires, and ε is the technology constant depended on the material of wires. In the following, we will assume ε and A to be the same for all wires. It means that the wire geometry is fixed and only d is modifiable.

Under the above formula (7), we can widen the distance between two adjacent pairs so that the effects of coupling capacitances can be reduced. Our spacing architecture is illustrated in Figure 3.5, where d_{min} is original (minimal) distance between any two adjacent lines and α is the distance ratio. The value of α is larger than or equal to 0 and stands for widened distance is larger than that of original distance (α+1) times. Because d_{min} is limited by technology of fabrication, narrowing the distance between two adjacent lines is impracticable. As the distance ratio α grows, although the number of coupling transitions is not changed, the smaller coupling capacitance implies less power dissipation. Nevertheless, widening the distance is costly in area and leads to increase the cost of the production, but it is a tradeoff between performance and cost. However, Spacing mechanism indeed resolves the coupling transitions between two adjacent pairs. In our experiment, we will apply different distance ratio α to observe the impact of them.

So far, we have introduced the essence of our low-power bus model, including ADEM, ADEM with reduced overhead, and Spacing. In the next chapter, we will evaluate the performance of our proposed methods and compare to other methods.

Chapter 4 Experimental Results

In this chapter, we will perform a number of simulations to evaluate the performances of our integrated method presented above. The goals of our simulations are as follows. First, we compare the overall performance among ADEM, ADEM_2L, and ADEM_4L. Next we compare the performance of our proposed methods to OEBI [9] and CBBI [13] with different measurements of capacitance ratio, distance ratio, and bus width. Finally, we discuss the overall overhead of our integrated method and compare to those of others based on delay time and circuit area.

4.1. Overview of simulation [9, 13, 18]

In simulation environment, we will simply calculate the number of self- and coupling transitions on all the capacitances. The overall power dissipation for transmission of data streams can be calculated by using the following formula (i.e. combining formula (1) and (2)):

$$\sum_{bus\ cycles} P_d = \sum_{bus\ cycles}\left(\sum_{M} P_{ds} + \sum_{M} P_{dc}\right) = \sum_{bus\ cycles}\left(\sum_{M}\frac{1}{2}\alpha_L C_L V_{dd}^2 + \sum_{M}\frac{1}{2}\alpha_C C_C V_{dd}^2\right) \qquad (8)$$

In the formula, the number of self-transitions (α_L) and coupling transitions (α_C) affect the total power dissipation while the other parameters are considered as constant. Furthermore, the capacitance ratio (λ) will be changed between different degrees of fabrication. The power dissipation between different degrees of fabrication can be calculated by replacing C_c with $\lambda \cdot C_L$, and it can be estimated as:

$$\sum_{bus\ cycles} P_d = \sum_{bus\ cycles}\sum_{M}\left(\frac{1}{2}(\alpha_L + \lambda \cdot \alpha_C) C_L V_{dd}^2\right) \qquad (9)$$

Then, Spacing mechanism is used to handle the pair transitions between each pair of lines. The value of coupling capacitances will be changed due to different distance

Parameter	Explanation	Values
λ	Capacitance ratio (C_C/C_L)	{3.9, 5.4, 7.4, ∞}
α	Distance ratio (d=d_{min}(1+α))	{0, 1/3, 2/3, 1, 4/3, 5/3, 2, 7/3, 8/3, 3}
M	Bus width	{8, 16, 24, 32, 40, 48, 56, 64}

Table 4.1. Parameters in our experiment.

between each pair of lines. In our experiments, the power dissipation with a certain length of distance can be calculated by replacing C_c with formula (7), and which can be estimated as follows:

$$\sum_{bus\ cycles} P_d = \sum_{bus\ cycles}\left(\sum_{M}(\frac{1}{2}\alpha_L C_L V_{dd}^2) + \sum_{M}(\frac{1}{2}\alpha_C(\varepsilon\cdot\frac{A}{d_{min}(1+\alpha)})V_{dd}^2)\right) \qquad (10)$$

By the calculation of these formulas (8~10), we can evaluate the reduction of power dissipation by our proposed integral methods.

There are three parameters in these formulas, i.e. the capacitance ratio (λ), the distance ratio (α), and bus width (M). Table 4.1 shows the given values of them used in our experiments. The capacitance ratios (λ) are set to 3.9, 5.4, and 7.4 for 90 *nm*, 65 *nm*, and 55 *nm* technologies, respectively. The distance ratio (α) is assumed for the distance is four times larger than that of original at most. However, there are still several parameters not given, including C_L, C_C, V_{dd}^2, A, and ε. They are considered as constants in our experiments and comparisons. The benchmarks used in our experiments are the multimedia files because they are common used in handheld devices. There are also no accredited benchmarks for these benchmark files, thus they are chosen arbitrarily.

4.2. Results analysis

In this section we first evaluate ADEM, ADEM_2L, and ADEM_4L and observe the results of performance and overhead. Meanwhile, we arbitrarily pick out ten files for each type of benchmark file. The data in the graph is the average of the simulations of the ten benchmark files. The performance is measured by using average power saving which is defined as [12]:

$$Average\_power\_saving = \left(1 - \frac{\sum_{bus\ cycles} P_d(encoded)}{\sum_{bus\ cycles} P_d(original)}\right) \times 100\% \qquad (11)$$

Then we will compare our integrated methods to OEBI and CBBI under various capacitance ratio (λ), distance ratio (α), and bus width (M).

4.2.1. Power dissipation caused by informed lines

Figure 4.1 presents the average power saving for ADEM, ADEM_2L, and ADEM_4L. Figure 4.2 presents the average power saving without considering power dissipation caused by informed lines. In these two figures, ADEM which considers all the types of pairs gets the best results in all benchmark files, but five informed lines counteract the achieved average power saving. In contrast, although ADEM_2L gets almost only half average power saving compared to ADEM, the two additional informed lines counteract only a little average power saving. Overall, ADEM_4L presents the best trade-off between power dissipation caused by bus lines and by informed lines. In the later figures, we will show the average performance of these ten benchmark files with considering power dissipation caused by informed lines.

4.2.2. The impact of capacitance ratio, distance ratio, and bus width

In the following graphs, the difference of performance between our methods and OEBI is the effect of the number of encoding methods just in a bus cycle. Our ADEM use various encoding methods in a bus cycle while OEBI uses only one. The difference between our methods and CBBI is that CBBI only decreases the number of toggling events. The average power saving under various capacitance ratios (λ), distance ratios

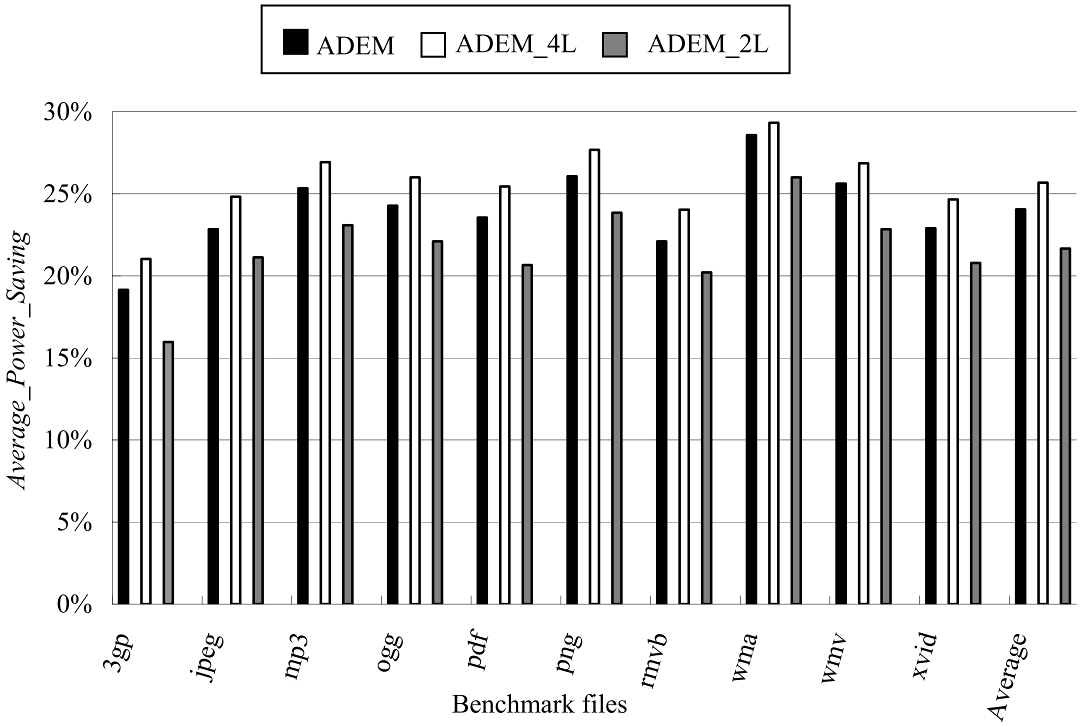


Figure 4.1. The average power saving of ADEM, ADEM_2L, and ADEM_4L (λ=3.9, α=0, M=32).

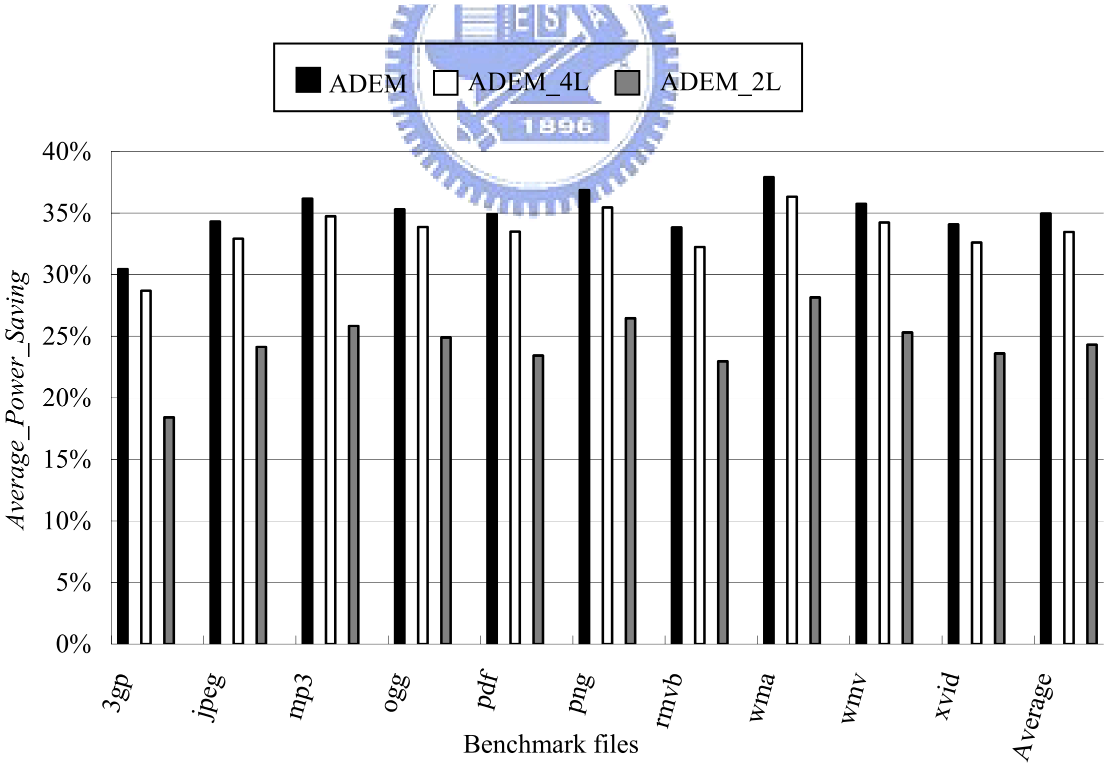


Figure 4.2. The average power saving of ADEM, ADEM_2L, and ADEM_4L without considering informed lines (λ=3.9, α=0, M=32).

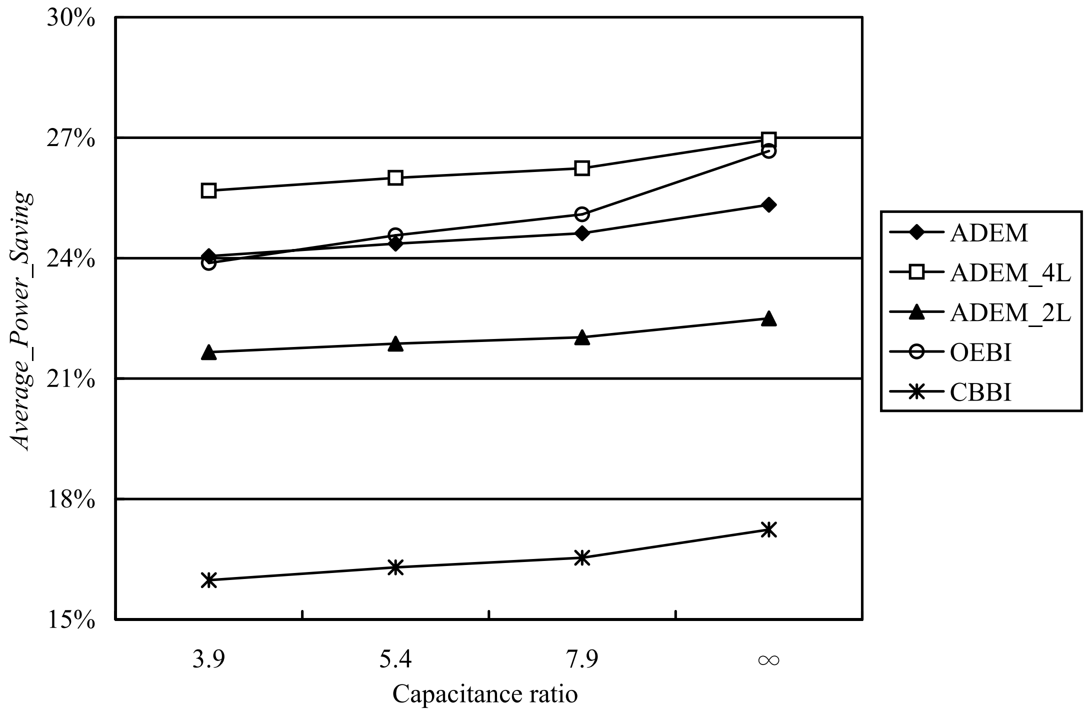


Figure 4.3. The average power saving under various capacitance ratios (α=0, M=32).

(α), and bus width (M) will be described below.

Figure 4.3 presents the average power saving including power dissipation caused by informed lines under various capacitance ratios (λ). While λ grows, power dissipation caused by coupling transitions also grows. Thus, all the methods which consider coupling transitions will show better results under larger λ. In our proposed methods, ADEM considers all types of pair while ADEM_4L and ADEM_2L do not, thus the growth of trend for ADEM is a little better than ADEM_4L and ADEM_2L. Then, the difference between our methods and those of others is that we don't consider coupling transitions between each pair in this moment. OEBI shows the best growth of the trend because all coupling transitions are considered. CBBI takes into account only toggling events instead of all coupling transitions, thus the growth of the trend is not as good as OEBI. In any case, our ADEM_4L still shows the best result under various λ ratios.

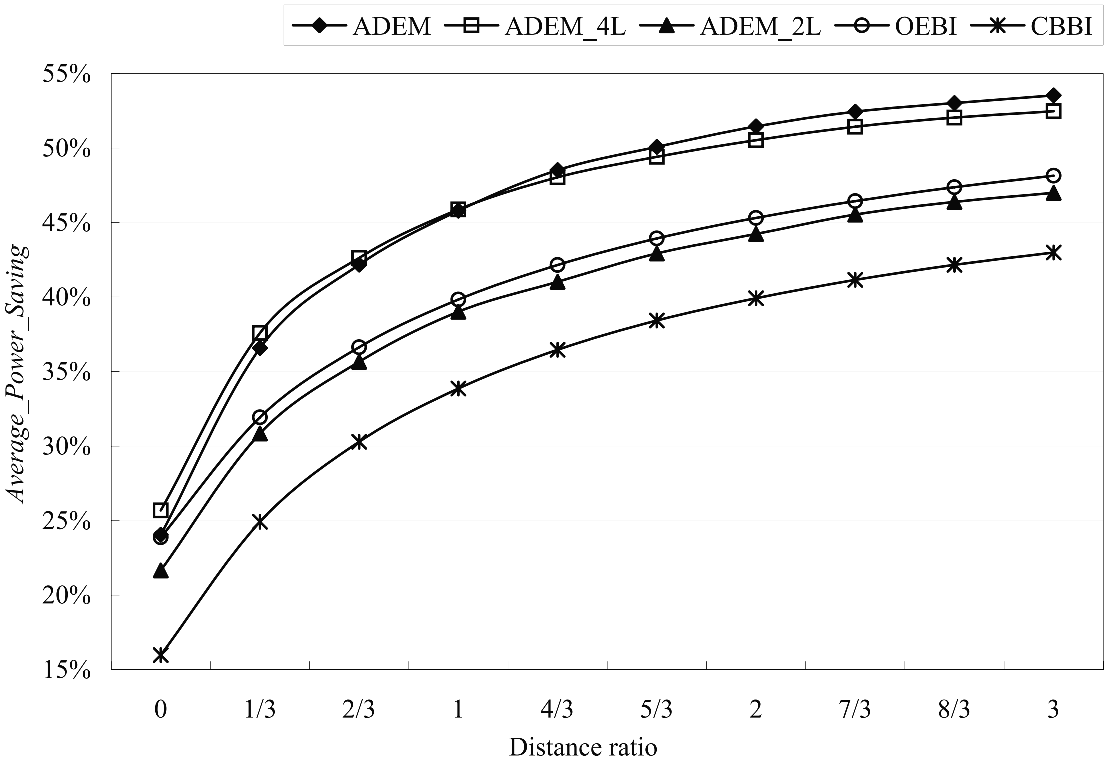
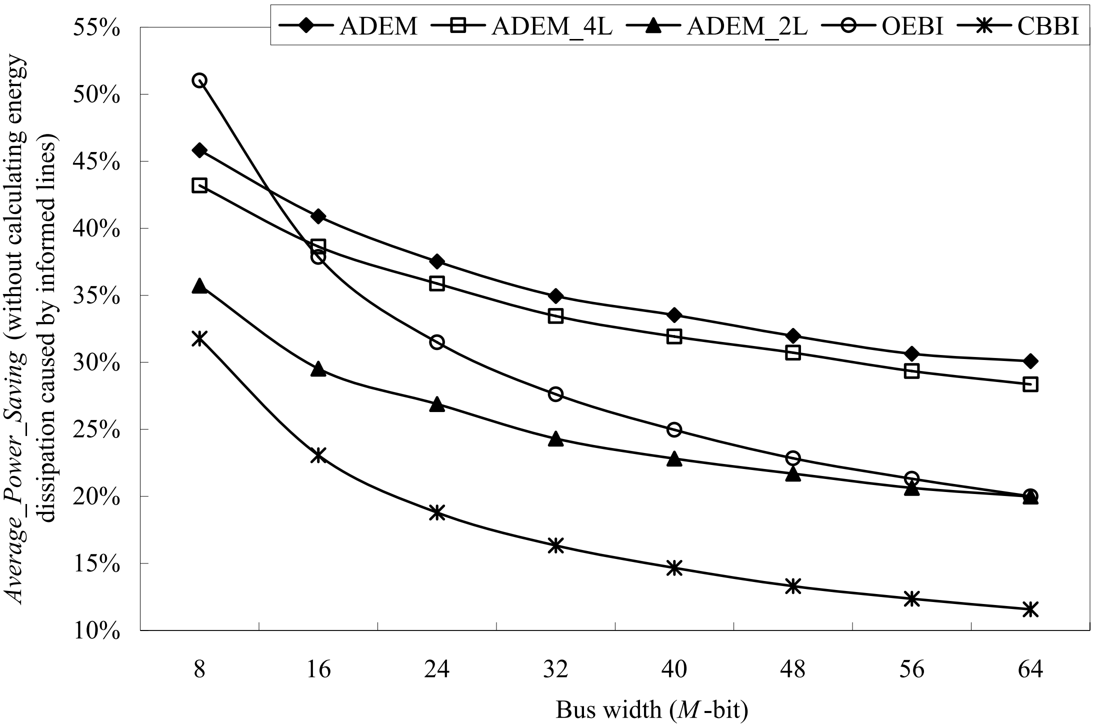


Figure 4.4. The impact of Spacing mechanism (λ=3.9, M=32).

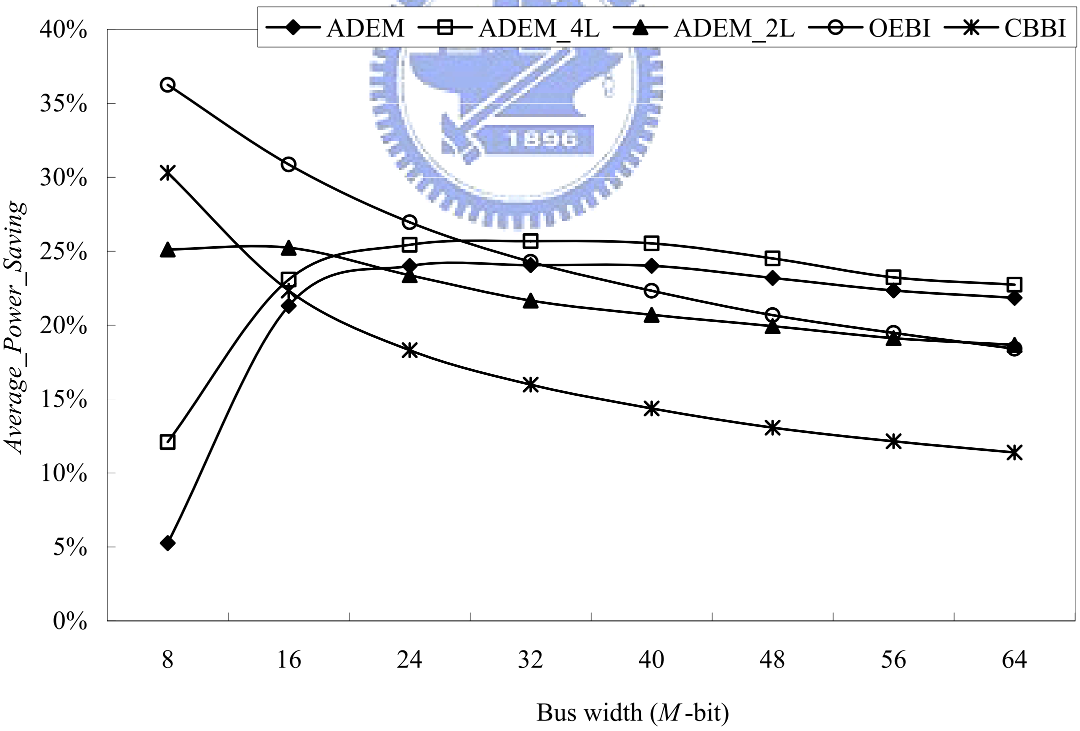
Figure 4.4 presents the impact of Spacing mechanism. Spacing improves more average power saving for all these methods. Because our methods doesn't consider the coupling transitions between each pair of lines, the power dissipation caused by these coupling transitions will occupy large portion of total power dissipation. Hence, the gap of average power saving between our methods and OEBI turns into larger in widener distance. Also, in the range from α=0 to 2/3 Spacing improves our methods more than that of OEBI and CBBI. While α is larger than 1, we find that the performance of ADEM shows better result than that of ADEM_4L. The reason is that the pair transitions here occupy larger portion of total power dissipation than that caused by coupling transitions between each pair of lines. Meanwhile, ADEM considers all types of pair while ADEM_4L doesn't. In summary, Spacing does save much power for all encoding methods, and especially for our methods. However the width of distance is a tradeoff between the amount of average power saving and size of circuit area.

Figure 4.5(a) and (b) present the effect without and with considering power dissipation caused by informed lines under various bus widths, respectively. The major difference between our methods and others is the number of encoding methods applied in a bus cycle. Without considering informed lines, the bus width affects OEBI and CBBI more than our methods. The reason for this effect is that OEBI and CBBI use only one encoding method in a bus cycle, and that can't cover so many cases of coupling transition in widened bus system. In contrast, we apply various encoding method instead of one in a bus cycle, which can cover more cases of coupling transition in widened bus system. Thus, the influence of widened bus system is less than those of OEBI and CBBI. Besides, if we take informed lines into consideration, ADEM and ADEM_4L will show only a little improvement in the case of M=8. The reason is that the power dissipation caused by 4 and 5 affects average power saving more than that of other cases of M. While M becomes larger, the trend turns into more stable because the power dissipation caused by informed lines here doesn't occupy so large portion of total power dissipation. In summary, our ADEM and ADEM_4L will show better result in the case of $M \geq 32$.

So far, we have evaluated the performance of our methods. We showed that ADEM_4L presents the best result in most cases. We also found that Spacing mechanism does improve bus encoding techniques a lot. As the fabrication trends to advancement, ADEM_4L also trends to save more power dissipation. Only in the case of smaller bus width, i.e. $M < 32$, the achieved performance will be counteracted due to 4 additional informed lines, and doesn't show the best result. However, while M becomes larger, ADEM and ADEM_4L still exhibits better result than those of others.



(a)



(b)

Figure 4.5. The impact of bus width (a) without (b) with considering power dissipation caused by informed lines (λ=3.9, α=0).

4.3. Overhead comparison

To realize our proposed integrated method, we first need to implement a decision circuit in the encoder which recognizes the appearance number for each type of pairs. Then, a set of combinational logic are used to encode each type of pair with corresponding encoding method. In the decoder, the informed lines control the decoding circuit applying different decoding methods. Thus, the overhead of circuit area contains those of decision, encoding, decoding circuit, and informed lines. The major overhead of delay time will be limited in the decision circuit because the comparator requires more time. In addition, a little power will be dissipated due to additional inserted circuits. It can be negligible because the power dissipation caused by inserted circuit is much less than the transitions on the capacitances [9].

In OEBI, there are only a few invert gates used in encoding and decoding circuit. Thus, they don't cost much delay time and circuit area. Nevertheless, OEBI should choose the best candidate with the least coupling transitions to be the codeword. The decision circuit needs to calculate the number of coupling transitions for each candidate and then compare. In contrast, our methods only recognize the appearance number for each type of pair. For the size of decision circuit, ours occupies less than and nearly half compared to OEBI. For the delay time of decision circuit, OEBI requires more time to pick the codeword out from the 4 candidates. Overall, the area of encoder in our methods is smaller than that in OEBI, but the decoder is larger than that in OEBI.

In CBBI, only the toggling events are taken into consideration instead of all coupling transitions. An inverter in decision circuit is used to compare the number of *penalties* and *rewards*. It only costs a little circuit area and delay time. However, although the overall overhead in CBBI is less than our methods, the experiment results show that our methods present better performance than CBBI.

		ADEM	OEBI [9]	CBBI [13]
Circuit area	Decision	4 counters and 1 comparator	4 coupling counter and 1 adder	An inverter
	Encoding and decoding	A set of combinational logic	A little invert gates	A little invert gates
Delay time		Recognizing the types of pair *A*, *B*, *C*, and *D*	Calculating the number of coupling transitions	Comparing the number of *penalty* and *reward*
Informed lines		5, 4, or 2	2	1

Table 4.2. The comparison of circuit area, delay time, and the number of informed lines for ADEM, OEBI, and CBBI.

Besides, for the number of informed lines, our methods require 2, 4, or 5 lines for recovery but only 2 in OEBI and 1 in CBBI. These informed lines will increase not only circuit area but also the cost of production. They also introduce additional power dissipation. However, inserting the informed lines is unavoidable overhead for common encoding methods focused on low-power. Although our methods require more additional informed lines, we still achieve the goal of low-power dissipation.

Table 4.2 lists the comparison of circuit area, delay time, and the number of informed lines for these three methods. In the circuit area, the required components of circuit used in decision, encoding and decoding are listed. In the delay time, only the critical function which costs the most time is compared. Overall, the circuit area and delay time of ADEM is less than OEBI, but more than CBBI. In the next chapter, some conclusions and future work are remarked.

Chapter 5 Conclusions and Future Work

Finally, we will conclude our method and give some future work in this chapter.

5.1. Conclusions

In this thesis, we propose a method named *amount-driven encoding method* (ADEM) for low-power on-chip data bus design. It reduces the number of self- and coupling transitions by using encoding technique and decreases the value of coupling capacitance by applying Spacing. In summary, ADEM has the following features and contributions.

(1) In order to avoid the influence of neighbor coupling transitions, ADEM considers the coupling transitions in two phases. In phase one while ADEM encodes the pairs, it first ignores the power dissipation caused by the coupling transitions between each pair of lines. Then in phase two, the unresolved coupling transitions between each pair will be dealt with Spacing. This two phases design avoids the influence upon two adjacent coupling capacitances, thus our encoding method design can only concentrate on the improvement of pair transitions. Therefore, for the power dissipation caused by pair transition, ADEM save more power than those of OEBI and CBBI.

(2) Through our bus encoding method design, ADEM separates all the pairs into four types. Each type of pair will be encoded with four encoding methods. During encoding, ADEM first recognizes the order of appearance number for each type to process encoding without calculating total power dissipation. The experiment result shows that ADEM which applies four encoding methods outperforms only one in OEBI and CBBI. Moreover, because the order of the appearance number is independent of the

bus width, ADEM is well suited for high-performance devices which contain wider bus system. Even as the fabrication trends advancing, ADEM saves more power and is still the most effective method compared to OEBI and CBBI.

(3) The overhead of delay time and circuit complexity is less than that of OEBI. The reason is that ADEM can encode by recognizing the appearance number of each type of pair instead of calculating total power dissipation. Meanwhile, the coupling transition between each pair of lines can be ignored during encoding.

5.2. Future work

In addition to our previous features, there are still some attractive issues worthy of further investigations. First, ADEM is mainly designed for data buses. It may not perform well on instruction or address buses, because it cannot exploit the high correlation of instruction or address data streams. In the future, we will try to find the un-correlated part of instruction or address stream, and then modify ADEM applying to the part. For instance, the un-significant part of address stream almost has weak correlation, so that we can apply ADEM to the un-significant part of address stream.

Second, we assume that the signals on the all lines are synchronized, i.e. there is no delay skew between them. However, for on-chip bus, a relative delay between data on neighbor lines usually occurs due to process, voltage and temperature variation. Therefore, delay skew between neighbor lines may change the charge, discharge, or toggle events of coupling capacitance [9, 19]. For example, the toggling event 10→01 will become two separate events 10→00 followed by 00→01, so that we can't exactly estimate the number of coupling transitions by our proposed method. In the future, we may consider that the signals with delay skew of normal distribution, and then re-estimate the number of self- and coupling transitions.

Finally, it should be noted that our proposed method doesn’t consider the coupling transitions between nonadjacent lines, because its effects on power dissipation seems relatively small. However, in an ultra deep-submicron technology design in the future, the events of coupling transition is affected not only by two adjacent lines but also by two nonadjacent lines. In the case, even the coupling transition is not affected by adjacent lines instead of nonadjacent lines, we also need to estimate it

Bibliography

[1] P. P. Sotiriadis and A. Chandrakasan, “Bus energy minimization by transition pattern coding (TPC) in deep sub-micron technologies,” *Proc. of 2000 IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 322–327. 2000.

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

[3] Y. Shin, S. Chae, and K. Choi, “Partial bus-invert coding for power optimization of application-specific systems”, *IEEE Trans. on VLSI Systems*, Volume 9, pp. 377–383, Apr. 2001.

[4] L. Benini, G. De Micheli, A. Macii, E. Macii, and M. Poncino, “Reducing power consumption of dedicated processors through instruction set encoding”, *Proc. of IEEE 8th Great Lakes Symposium on VLSI*, pp.8–12, Feb. 1998.

[5] 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”, *Proc. of IEEE 7th Great Lakes Symposium on VLSI*, pp. 77–82, Mar. 1997.

[6] Y. Aghaghiri, F. Fallah, and M. Pedram, “Irredundant address bus encoding for low-power”, *Proc. of IEEE International Symposium on Low-Power Electronics and Design*, pp. 182–187, Aug. 2001.

[7] W. F. Yang, “Instruction level scheduling for low-power on VLIW DSP”, *Master Thesis, National Chiao-Tung University*, June 2005.

[8] T. Lindkvist, J. Löfvenberg, H. Ohlsson, K. Johansson, and L. Wanhammar, “A power-efficient, low-complexity, memoryless coding scheme for buses with dominating inter-wire capacitances”, *Proc. of the 4th IEEE international Workshop on System-on-Chip for Real-Time Application*, Jul. 2004.

[9] Y. Zhang, J. Lach, K. Skadron, and M. R. Stan, “Odd/even bus invert with two-phase transfer for busses with coupling”, *Proc. of IEEE International Symposium on Low-Power Electronics and Design*, pp. 754–757, Aug. 2002.

[10] P. Petrov and A. Orailoglu, “Low-power instruction bus encoding for embedded processors”, *IEEE Trans. on VLSI Systems*, Volume 12, Issue 8, pp. 812-826, Jul. 2004.

[11] S. K. Wong and C. Y. Tsui, “Re-configurable bus encoding scheme for reducing power consumption of the cross coupling capacitance for deep sub-micron instruction bus”, *Proc. of IEEE Annual Symposium on VLSI*, pp. 167-172, Feb. 2003.

[12] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, “A dictionary-based en/decoding scheme for low-power data buses”, *IEEE Trans. on VLSI systems*, Volume 11, No. 5, Oct. 2003.

[13] M. Ghoneima and Y. Ismail, “Low Power Coupling-Based Encoding for On-Chip Buses”, *Proc. of the 2004 Int. Symposium on ISCAS’04*, Volume 2, pp. 23-26, May. 2004.

[14] R. Arunachalam, E. Acar, and S. Nassif, “Optimal shielding/spacing metrics for low power design”, *Proc. of IEEE Annual Symposium on VLSI*, pp. 167-172, Feb. 2003.

[15] E. Musoll, T. Lang, and J. Cortadella, “Working-Zone encoding for reducing the energy in microprocessor address buses,” *IEEE Trans. on VLSI System*, Volume 6, pp. 568–572, Dec. 1998.

[16] R. B. Lin, “Coupling reduction analysis of bus-invert coding”, *IEEE Symposium on Circuit and System 2005, ISCAS’05*, Volume 6, pp. 23-26, May. 2005.

[17] Enrico M., Massimo P., Sabino S., “Combining wire swapping and spacing for low-power deep-submicron buses”, *Proc. of .the 13th ACM Great Lakes symposium on VLSI*, pp. 198-202, 2003.

[18] S. J. Ruan, Edwin N., Uwe S., “Simultaneous wire permutation, inversion, and spacing with genetic algorithm for energy-efficient bus design”, *Proc. of the 19th IEEE Symposium on IPDPS'05*, Volume 12, pp. 233.1, 2005.

[19] M. Ghoneima and Y. I. Ismail, “Delayed line bus scheme: a low-power bus scheme for coupled on-chip buses”, *Proc. of the 2004 International Symposium on Low Power Electronics and Design, ISLPED ’04*, pp. 66-69, 2004.