Dynamic Bus Encoding based on Realistic RLC Table with Low Power Consideration

by

Ke-Yu Chen

Department of Computer Science and Information Engineering
National Taiwan University

Advisor

Prof. Feipei Lai

June 2004

Contents

List of Figures

List of Tables

Abstract

In the years to come, interconnect will be a challenge to surmount for deep sub-micron (DSM) technologies. The use of DSM technology increases the capacitive and inductive coupling which leads to severe crosstalk noise, more power dissipation and malfunction of the chip between neighboring wires. We propose a method to reduce crosstalk noise on buses based on dynamic coding scheme. The proposed method deals with capacitive and inductive effects at the same time by using the realistic RLC table and the segmented bus-invert method. The experimental results show that our approach reduces bus power consumption up to 7%.

Chapter 1 Introduction

As G. Moore's law predicted [1], we have made extraordinary progress in computer performance during the last three decades. To keep on advance at the same rate, there are some major technological problems to be settled.

In the years to come, transistors are expected to become even smaller and faster. It is a momentous subject that how to interconnect them efficiently. The ***interconnect*** means that the set of wires in the circuit responsible for communication, clock distribution which causes power distribution.

In modern deep sub-micron (DSM) technologies, the communication problem has become harsh and complex. The reasons for this are the following:

1. **Increased coupling between the lines of buses** [5, 6, 7].

   To maintain a linear resistance of reasonable size, it is necessary for wires with the higher aspect ratio (height/width) and the smaller distances. This decreases the resistance of the wires, hence making the wire impedance due to inductance comparable to that due to resistance, and inductance effects become more and more significant.

2. **The distributed behavior of thin and long lines.**

   With the increase of chip size, it is fairly typical that many wires are long and run in parallel, which increases the inductive cross-talks between them. Hence, longer wires cause the more significant coupling effect.

3. **Increased speed of microprocessor's core, the communication rate between components must be raised as well.**

   Higher clock frequency will induce more inductive noise because of the faster changing rate of the current.

The issues above are expected to become even more dominant in the future technologies [6, 8]. Interconnect will be a major design problem for the next two decades [2, 3].

It is interesting that all of these problems, coupling, distributed phenomena, or faster clock frequency did not occur in earlier technologies. This means that early approaches to estimate power and delay in buses may not be suitable for current DSM technologies. Moreover, techniques invented to reduce power consumption do not perform well in

modern technologies.

A large number of studies have been made on low power bus design. However, many of them focus on reducing bus switch activity of each signal line [10, 11, 12, 13], and do not take coupling effects into account. Hence, they are not directly related to crosstalk issues. Furthermore, some of these methods are only applicable to address buses [11, 14], which is not fit for other designs.

Basically crosstalk is caused by capacitance and inductance effects. Some approaches address coupling capacitance at physical design level by optimizing routing and gate/wire sizes [15, 16]. In [16, 17], similar approaches are introduced to find the permutation for the bus lines to minimize coupling capacitance effects. These approaches are based on profiling the bus line activity and are suited to special embedded system design; however they can not dynamically adjust to current signal on the bus.

The other way to reduce crosstalk effect is to minimize the coupling inductance. The inductive effect impacts signal integrity and performance, and is recognized as significant. Several techniques have been proposed to deal with this effect. [18] proposed a shielding technique, in which the signal lines are inter-digitated with Vdd or Ground alternatively. This approach is to isolate signal lines from their neighbors, and can be used to minimize coupling capacitive and inductive noises when combined with ordering nets [19]. Dedicated ground planes is another useful method to reduce inductive effect where the layers above and below the signal lines are dedicated to Vdd or Ground [20, 21]. These techniques focus on layout topology and need more area overhead and power consumption.

In the thesis, we propose a new dynamic bus encoding scheme for reducing bus power consumption, called ***DBE-RLCT*** *(Dynamic Bus Encoding with RLC Table)*. The proposed approach characterized by using a realistic RLC table considers the coupling capacitive and inductive effect at the same time. Cooperating with segment data inversion, partial data bits might be inverted to reduce crosstalk effect.

Chapter 2 Background

On-chip and inter-chip communication is responsible for a significant amount of power dissipated in modern digital circuits [6, 9]. Buses are the circuit units realizing this communication. Reliable estimation of the power that buses draw from the power supply is important. In this chapter, we introduce some mechanisms that cause crosstalk and how to construct basic transmission line structures typically used in digital systems.

2.1 Capacitive Effects

Usually, modern SoC are equipped with multiple processing units, memories and functional components to achieve the goal of the SoC. These components communicate with each other via various buses: data bus, instruction bus, and address bus. Due to the switching activity on these buses, power is dissipated and noise is generated.

2.1.1 Self Capacitance

If the width of the wire is substantially larger than the thickness of the insulating material as shown in Figure 2-1, it may be assumed that the electrical-field lines are orthogonal to the capacitor plates, and that its capacitance can be modeled by the ***parallel-plate capacitor model***.

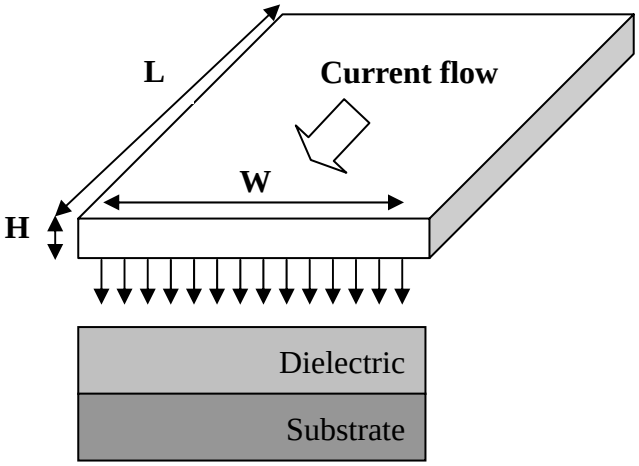


Figure 2-1: Parallel-plate capacitance model of interconnect wire.

Under those circumstances, the total capacitance of the wire can be approximated as

$$C_{\text{int}} = \frac{\varepsilon_{di}}{t_{di}} WL \tag{2.1.1}$$

where W and L are, respectively, the width and length of the wire, and t_{di} and ε_{di} represent the thickness of the dielectric layer and its permittivity. SiO_2 is the dielectric material of choice in integrated circuits, although some materials with lower permittivity, and thus lower capacitance, are coming into use. ε is typically expressed as the product of two terms, this is

$$\varepsilon = \varepsilon_r \varepsilon_0 \tag{2.1.2}$$

$$\varepsilon_0 = 8.854 \times 10^{-12} F / m$$

where ε_0 the permittivity of is free space, and ε_r is the relative permittivity of the insulating material. Table 2-1 presents the relative permittivity of several dielectrics used in integrated circuits.

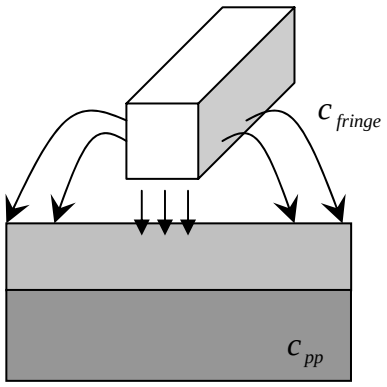
Table 2-1 Relative permittivity of some typical dielectric materials.

Material	ε_r
Free space	1
Aerogels	~1.5
Polyimides (organic)	3-4
Silicon dioxide	3.9
Glass-epoxy (PC board)	5
Silicon Nitride (Si_3N_4)	7.5
Alumina (package)	9.5
Silicon	11.7

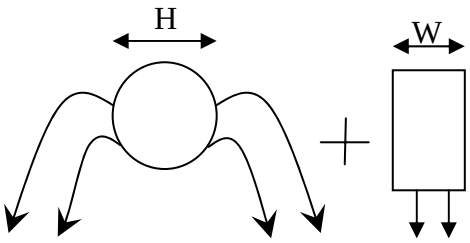
Over the years, we have witnessed a steady reduction in the W/H ratio such that it has dropped below unity in advanced processes. Under those circumstances, the parallel-plate model assumed earlier becomes inaccurate. The capacitance between the side walls of the wires and the substrate, called *fringing capacitance*, can no longer be ignored and contributes to the overall capacitance. This effect is illustrated in Figure 2-2(a). We use a simplified mode that approximates the capacitance as the sum of two components, just as shown in Fig 2-2(b). The resulting approximation, which is simple and works fairly well in practice, is

$$c_{wire} = c_{pp} + c_{fringe} = \frac{w\varepsilon_{di}}{t_{di}} + \frac{2\pi\varepsilon_{di}}{\log(t_{di}/H)} \qquad (2.1.3)$$

where $w = W - H/2$ is a good approximation for the width of the parallel-plate capacitor.



(a) Fringing fields



(b) Model for fringing-field capacitance

Figure 2-2: The fringing-field capacitance. The mode decomposes the capacitance into two contributions: a parallel-plate capacitance, and a fringing capacitance.

2.1.2 Coupling Capacitance

Today's processes offer many more layers of interconnect, which are packed quite densely in addition. In this scenario, the assumption that a wire is completely isolated from its surrounding structures and only capacitively coupled to ground becomes untenable. This is illustrated in Figure 2-3, where the capacitance components of a wire embedded in an interconnect hierarchy are identified. Each wire is not only coupled to the grounded substrate, but also to the neighboring wires on the same layer and on adjacent layers, resulting in mutual capacitance.

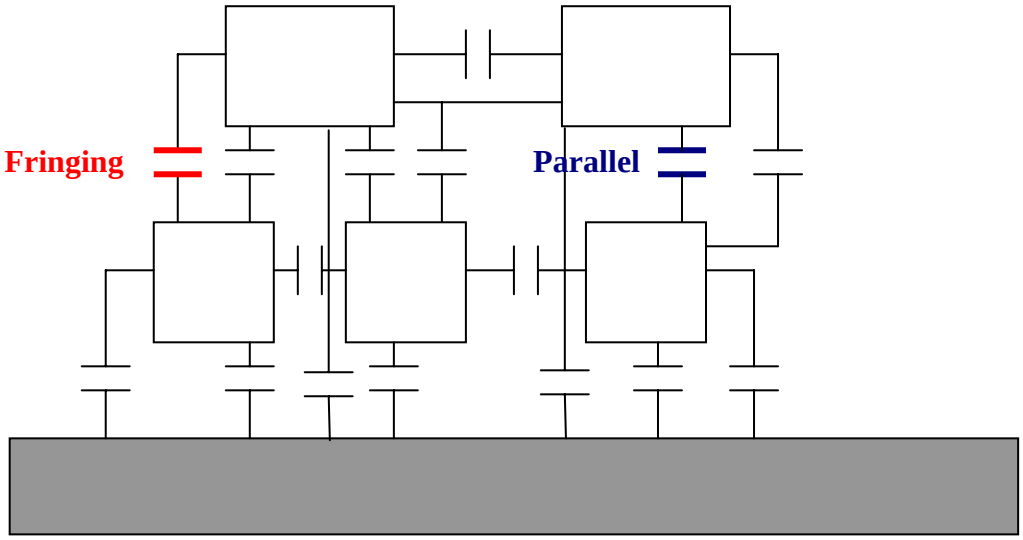


Figure 2-3: Capacitive coupling between wires in interconnect hierarchy.

Mutual capacitance will inject a current onto the victim line proportional to the rate in change of voltage on the driven line:

$$I_{noise,C_m} = C_m \frac{dV_{driver}}{dt} \tag{2.1.4}$$

Consider a pair of bus wires, during the operation there are three types of possible transitions when we just take dynamic charge distribution over coupling capacitances into account. Figure 2-4 (a) shows single line switching, where the coupling capacitance is charged (or discharged) with $C_x V$. In Figure 2-4 (b), the coupling capacitance will be

charged with $2C_xV$ if both wires transit into opposite directions. If both wires concurrently switch to the same voltage level, as shown in Figure 2-4(c), then the charge on the capacitance is not affected.

The crosstalk noise between neighboring signal wires changes the way of the signal propagation on some of the nets, slows down achievable clock speed, and brings about more power consumption.

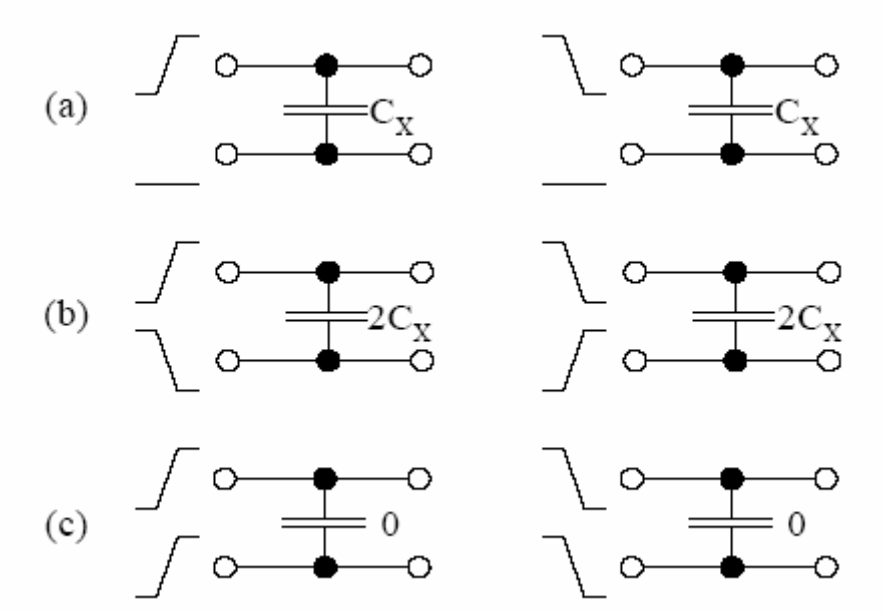


Figure 2-4: Line switching for two buses interconnects.

The crosstalk phenomenon may result from the existence of a capacitance between neighboring bus wires. With every single transition on a wire, noise is produced through this coupling capacitance on other wires.

2.2 Inductive Effects

Modern technology has demonstrated a relentless trend toward faster circuits, larger die sizes, shorter rise times, and smaller pulse widths. Circuit designers are designing chips operated by Gigahertz clocks on dies larger than $300mm^2$ using $0.25\mu m$ technology [22]. With these technology trends, on-chip inductance effects, such as delay slow-down, over-shoot, and inductive crosstalk, can no longer be ignored [23, 24].

Inductance effects have become more and more significant mainly because:

1. For performance considerations, some global signal and clock wires are routed with large widths and thicknesses at the top levels of the metal to minimize delays. This decreases the resistance of the wires, hence making the wire impedance due to inductance comparable to that due to resistance.

2. As the clock frequency increases and the rise time decreases, electrical signals comprise more and more high frequency components, making the inductance effects more significant.

3. With the increase of chip size, it is fairly typical that many wires are long and run in parallel, which increase the inductive crosstalk between them.

4. With the push of performance, some low resistivity metals have been explored to replace Al in order to minimize wire RC delay [25]. In [26], designers at Motorola demonstrated the success of using copper (Cu) wires. This could make the wire inductive reactance larger than the resistance.

Therefore, circuit designers have to consider on-chip inductance in order to ensure the functional correctness and performance of their high-speed designs.

2.2.1 Self Inductance

It has been defined in [33] that the self inductance of a wire is the number of field lines from its own current, per amp of current. This will be independent of the presence of another conductor’s current.

2.2.2 Partial Self Inductance

Of course, real currents flow in complete circuit flow in complete circuit loops. We have been looking at just a section of a wire. In this view, the only current that exists is the specific current in the section of wire. When we count the field lines, we have ignored the field lines from the current in the rest of the circuit. This type of inductance is called *the partial self-inductance* of the wire, to distinguish the fact that we are looking at only a part of the loop. In reality, you can never have a partial current - you must always have current loops, but the concept of partial inductance is a very powerful tool to understand and calculate the other flavors of inductance, especially if you do not know what the rest of the loop looks like yet.

In [32], Grover gave the following formula for estimating the partial-self-inductance of a straight wire with length, width and thickness *l, w, t* respectively.

$$L(nH) = 2l \times \left[\ln\left(\frac{2l}{w+t}\right) + 0.5 - k \right] \qquad (2.2.1)$$

In short, partial self-inductance is the self-inductance of a portion of a current loop with the inductive coupling from the rest return portion ignored.

2.2.3 Mutual Inductance

When we look at signals propagating in a transmission line, the current, at the wave front, is flowing though a loop composed of the partial self inductance of the signal path, and the partial self inductance of the return path. When current in one loop changes, it will change the number of field lines around the second loop and induce noise in the second loop.

Consider a typical on-chip topology including a signal net along with the power and ground grid in its vicinity. When the signal switches due to change in state of its driving gate, there are various currents that flow in the signal line and the neighboring grid. As shown in Figure 2-5 [34], the currents I_1 and I_2 form loops throughout the package and grid decoupling capacitances, while I_3 forms a current loop from the driver output, through the grid and back. When the current in loop changes, it changes magnetic field lines onto neighboring current loops, and induces mutual inductance to other wires.

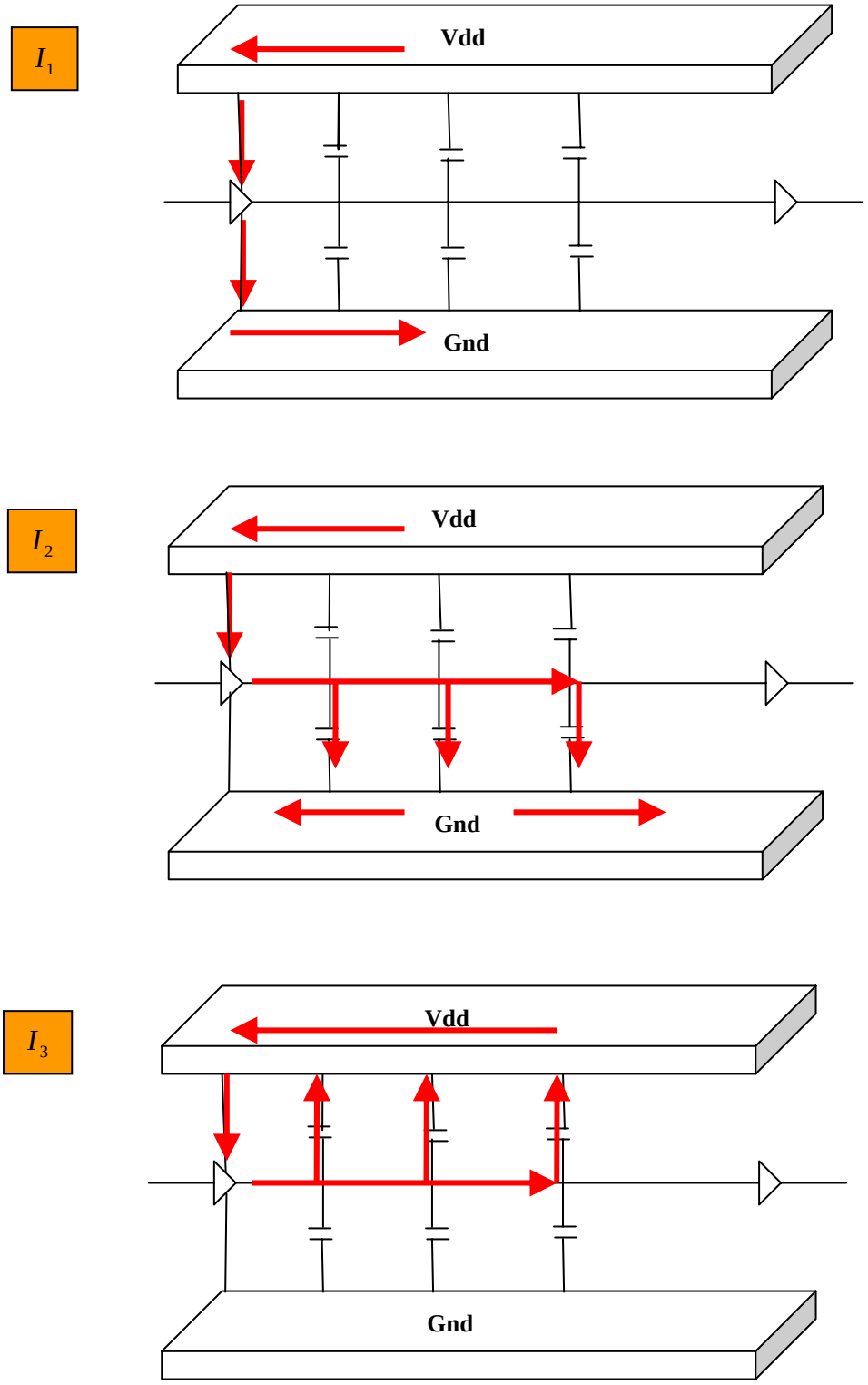


Figure 2-5: Current in Driver-Receiver-Grid topology.

Mutual inductance is one of the two mechanisms that cause crosstalk. Mutual inductance L_m induces current from a driven line onto a quiet line by means of the magnetic field. The coupling of current via the magnetic field is represented in the circuit model by a mutual inductance. Mutual inductance L_m will inject a voltage noise onto the victim proportional to the rate of change of the current on the driver line. The magnitude of this noise is calculated as

$$V_{noise,L_m} = L_m \frac{dI_{driver}}{dt} \tag{2.2.2}$$

Since the induced noise is proportional to the rate of change, mutual inductance becomes very significant in high-speed digital applications.

2.3 The RLC circuit for On-chip Bus Architecture

The use of Partial Equivalent Elements Circuit (PEEC) [35] method, based on partial inductance, has been proposed to construct the RLC model. A detailed PEEC-based model [36] can be constructed using interconnect resistance, inductance and capacitance, device decoupling capacitances, activity in the grid and pad/package inductance. Such a detailed PEEC model leads to a dense RLC circuit matrix requiring large SPICE simulation time. Hence, it must be combined with an acceleration technique like matrix sparification of reduced order modeling.

Just as shown in section 2.2, inductive effect must be taken into consideration in modern deep sub-micron technologies. To simplify and to keep accuracy of the RLC model at the same time, the wires have to be partitioned into sections. The coupling capacitors are inserted at section points. Figure 2-6 shows *N* equal-length sections. “L” means inductance, “C” means capacitance, and “R” means resistance. For example, “L12” means the mutual inductance between wire 1 and wire 2; “Cs2” means the self-capacitance of wire 2, and “C23” means the coupling capacitance between wire 2 and wire 3. To achieve enough accuracy, the length should not be too long. The *R* and *L* for each section are equal to the wires’ partial reactance divided by *N*. Notice that *L* is obtained from inductance-solver given the total length of the interconnect structure since *L* is not scalable. However, once *L* for a given total length is determined, we can section *L* by *N* for RLC netlist formulation. The self capacitance for each section is equal to the per-unit-length parasitic capacitances multiplied by the section length.

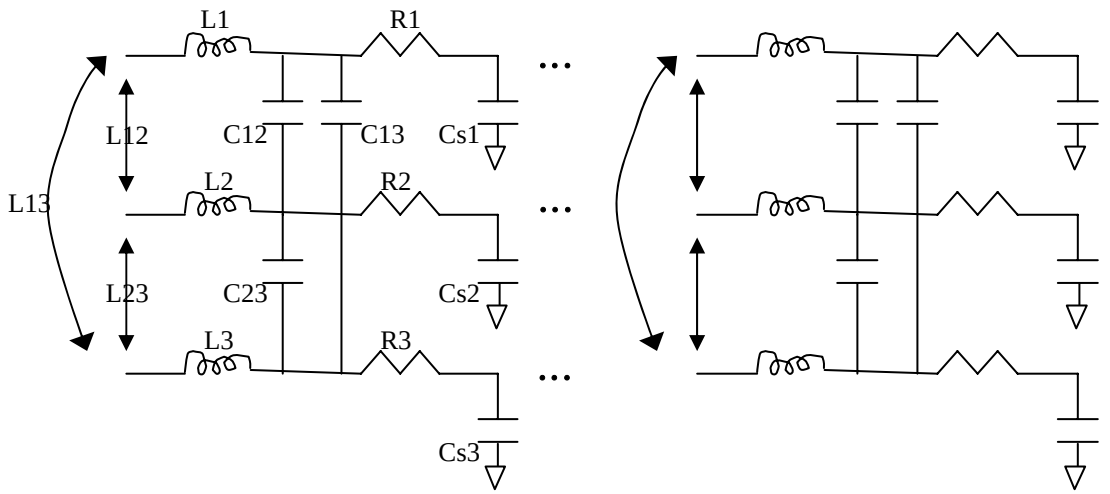


Figure 2-6: The equivalent RLC circuit for the 3-wire structure.

Equations (2.3.1) and (2.3.2) depict the typical method of representing the parasitic elements that govern the electrical performance of a coupled transmission line system. The inductance and capacitance matrices are known collectively as the ***transmission line matrices***.

$$\text{Inductance matrix} = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1N} \\ L_{21} & L_{22} & & \\ \vdots & & \ddots & \\ L_{N1} & & & L_{NN} \end{bmatrix} \qquad (2.3.1)$$

where L_{NN} is the self-inductance of line *N* and L_{MN} is the mutual inductance between lines *M* and *N*.

$$\text{Capacitance matrix} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1N} \\ C_{21} & C_{22} & & \\ \vdots & & \ddots & \\ C_{N1} & & & C_{NN} \end{bmatrix} \qquad (2.3.2)$$

where C_{NN} is the total capacitance seen by line *N*, which consists of conductor N's capacitance to ground plus all the mutual capacitance to other lines and C_{NM} means mutual capacitance between conductors *N* and *M*.

Chapter 3 Dynamic Bus Encoding Scheme

3.1 Motivation

In the past, we try to reduce the switching activities to achieve the goal of low power design, but the methodology is not suitable in modern design. In DSM technologies, inductive effect must be taken into account, and bus model has to be redefined as well. When it comes to inductance and capacitance, we have to discuss the RLC circuit firstly.

Figure 3-1 shows a typical RLC circuit, which contains inductance (L), resistance (R), capacitance (C), and DC. The equation of RLC loop can be represented as

$$L\frac{di}{dt} + Ri + \frac{1}{C}\int_0^t i \cdot dt + V_0 = v_g \qquad (3.1.1)$$

where $v(0) = V_0$, $i(0) = I_0$.

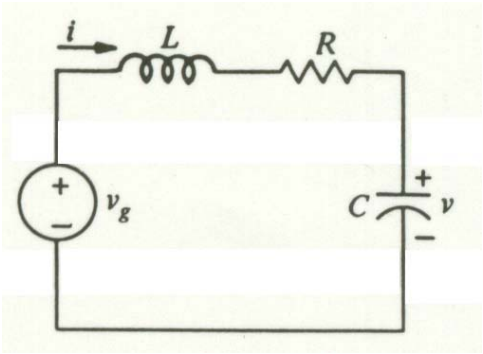


Figure 3-1: The RLC circuit

Because of the opposite phase of inductance and capacitance, reducing switching activities can only improve part of the equation 3.1.1. And the current flow is so unexpected that it is hard to find the regularity to improve capacitive and inductive coupling effect at the same time. Under such circumstance, we need to find the way to reduce power dissipation when both inductance and capacitance are taken into account.

3.2 DBE_RLCT Architecture

We propose a new dynamic bus encoding scheme, **DBE-RLCT** (Dynamic Bus Encoding with RLC Table), which is based on the methods of table look-up and segment data inversion. The **DBE-RLCT** can improve the capacitive and inductive coupling effect at the same time, using some simple combinational logic and some extra ROMs with small size.

The **DBE-RLCT** (Dynamic Bus Encoding with RLC Table) architecture consists of three primary components, *inversion-logic*, *decision-logic* and *RLC table*. We illustrate an example that the size of bus width is 32, as shown in Figure 3-2. Before the data transmits on the bus, we cut the data into divisions firstly. The *decision-logic* compares the current input data with the previous output data according to the realistic RLC table for each segment. Then the *decision-logic* generates a set of decision signals and sends them to the *inversion-logic*. The *inversion-logic* decides which part of the input data needs to be inverted according to the decision signals so that it can obtain a low crosstalk effect on the bus. Finally, the encoded data and the decision bits are put on the bus.

The architecture of the decoder is much simpler than that of the encoder. Only *inversion-logic* is required to convert the encoded data into the original data. The *inversion-logic* of the decoder inverts the data by the decision bits, which are generated by the DBE-RLCT encoder. It can be done simply by using some exclusive-or gates and in a very short time.

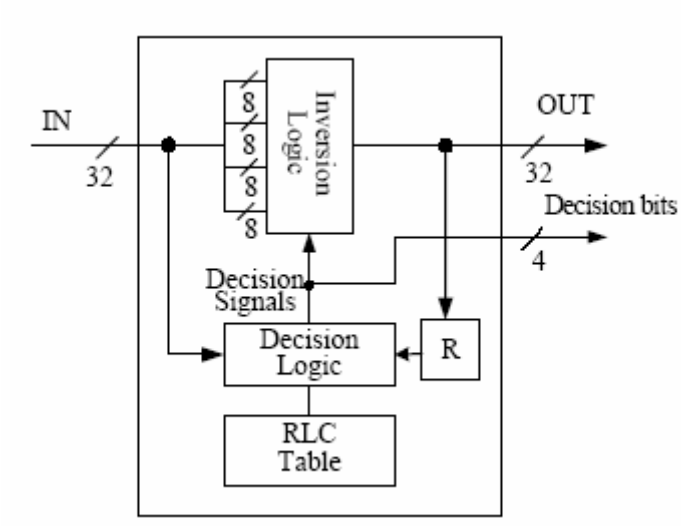


Figure 3-2: The DBE-RLCT encoder architecture.

Note that we partition the original input data by size of 8. In [28], it had been showed that smaller partition results in better improvement, but can only take you so far. According to the experimental result in [8], the best choice of partition size is 4 which brings about too many extra decision lines. To maintain the precision of experiments and to reduce the number of decision lines simultaneously, we use two RLC tables with size of 4 and an extra adder to sum up the weight of these two tables. By comparing the summed weight, the encoder can decide which partition should be inverted. The detailed architecture and operation of RLC table will be demonstrated in section 3.3.

3.3 Reduce Crosstalk Effect by RLC Table

The **DBE_RLCT** method reduces crosstalk effect by looking up the RLC table. First, when creating the RLC table, we build the bus model by writing SPICE netlist, and give some related parameters to generate the power value. Each record in the RLC table keeps track of the power value on the change of two consecutive transmissions. A bigger value indicates that there exists more serious crosstalk effect between bus wires. In the following, let us discuss segment data inversion using RLC table. In order to describe the DBE-RLCT method clearly, two definitions are made first:

Definition 1. *Assume the bus width is n, and partition them into m groups; n can be divided by m. That means there are* $(\frac{n}{m})$ *bits in each group. Let* G_k *as group k,* $G^i_{k\_ori}$ *as the original data of group k at* i^{th} *transmission,* $\overline{G^i_{k\_ori}}$ *as the inverse signal of* $G^i_{k\_ori}$*, and* $G^i_{k\_enc}$ *as the* i^{th} *encoded data of group k. Define* $V^i_k = G^{i-1}_{k\_enc} \phi G^i_{k\_ori}$*, where* ϕ *is the operation of calculating the power value when data switch from* $G^{i-1}_{k\_enc}$ *to* $G^i_{k\_ori}$*, and* V^i_k *is the delay between two times of transmissions. Likewise,* $\overline{V^i_k} = G^{i-1}_{k\_enc} \phi \overline{G^i_{k\_ori}}$ *is the power consumption when the data switch from* $G^{i-1}_{k\_enc}$ *to* $\overline{G^i_{k\_ori}}$.

Definition 2. *By definition 1, define*

$$DS_k^i = \begin{cases} 0, & if\ V_k^i \le \overline{V_k^i} \\ 1, & if\ V_k^i > \overline{V_k^i} \end{cases} \quad , and$$

$$\begin{aligned} MV_i &= MAX_{k=1,2...,m}\{MIN\{V_k^i,\ \overline{V_k^i}\}\} \\ &= MAX_{k=1,2,...,m}\{V_k^i + DS(\overline{V_k^i} - V_k^i)\} \end{aligned}$$

where DS_k^i *is the group k's decision bit at* i^{th} *transmission, and* MV_i *is the maximal power consumption value of the local minimum in each group at the* i^{th} *transmission.*

By definition 1 and definition 2, the process of the dynamic encoding method could be regarded as finding DS_k^i, $k = \{1, 2, ..., m\}$, such that SV_i is the minimum.

Figure 3-3 shows the inner structure of *decision-logic* and *inversion-logic*. In Figure 3-3(a), the DS_k is generated by comparator, which compares V_k with $\overline{V_k}$. When V_k is equal to or smaller than $\overline{V_k}$, DS_k is set as 0; in contrast, if V_k is larger than $\overline{V_k}$, DS_k is set as 1. The value of V_k and $\overline{V_k}$ are generated by estimators. The estimator's operation is to look up the RLC table, and return back the power consumption value when data transfer from $G_{k\_enc}^{i-1}$ to $\overline{G_{k\_ori}^i}$, or $\overline{G_{k\_ori}^i}$. The inversion-logic, shown in Figure 3-3(b), decides which group's data need to be inverted according to the decision signal, DS_k. If DS_k is 0, the multiplexer of the group *k* selects the original input data as the output data of the i^{th} transition; and if DS_k is 1, it selects the inverse data as the group k's output. Thus we can obtain a smaller crosstalk effect by inverting partial input data.

It must be noted that the inversion of partial input data is not for minimizing the hamming distance between two consecutive transmissions, which is proposed in [29]. It is used to select a better solution for crosstalk problem.

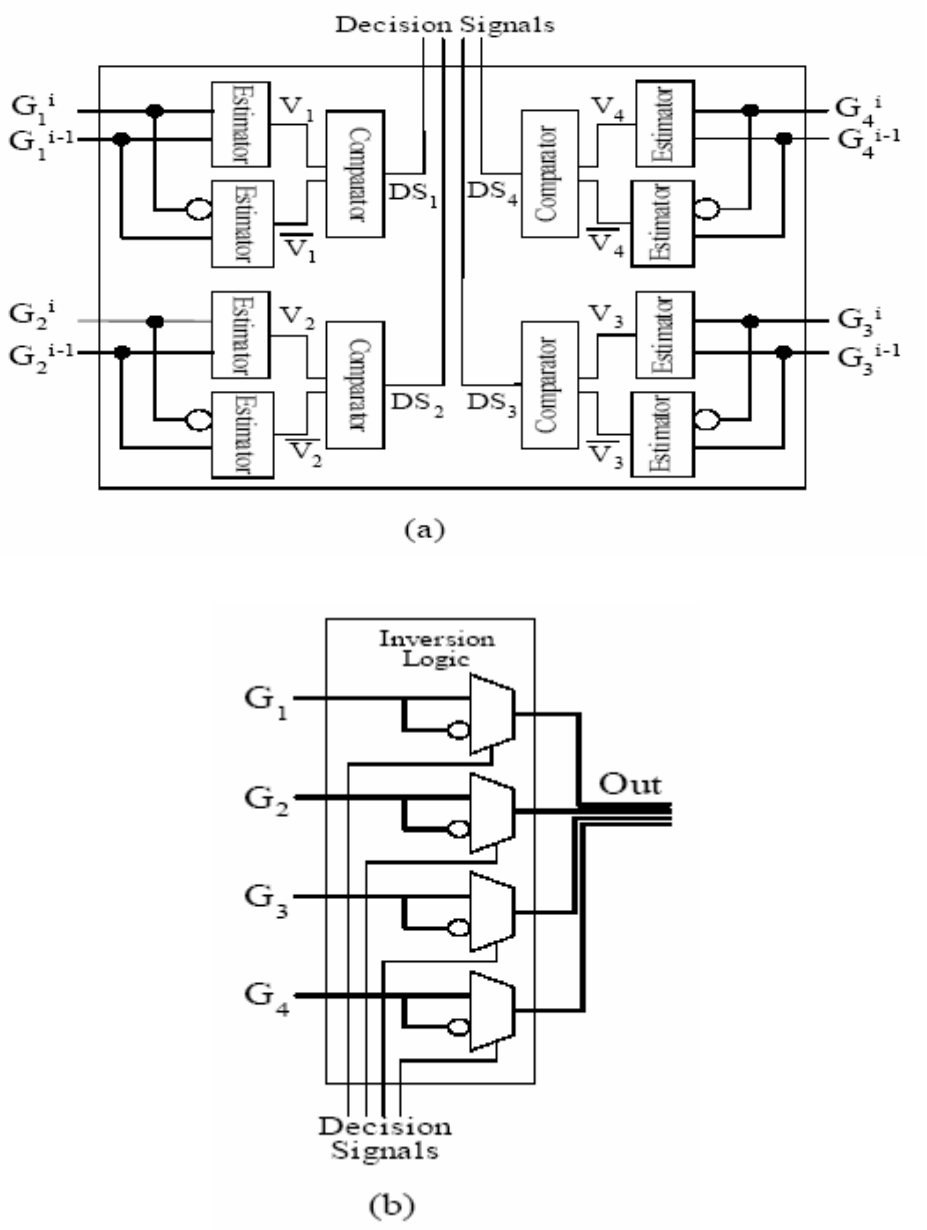


Figure 3-3: Inner structure of (a) decision-logic and (b) inversion-logic.

3.4 RLC Table

3.4.1 Architecture

The architecture of RLC table is shown in Table 3-4.

Table 3-1: RLC Table

Bus Transition Type	**Weight (Unit: nW)**
Ffrf	1.3
01ff	0.8
11ff	0.8
…	…
Rrrr	0.2

In Table 3-1, “Bus Transition Type” means the transition on the bus, and “Weight” means power consumption of the specific transition in the first column which was simulated by HSPICE. There are four notations to represent the transition, that is, *0, 1, f,* and *r*. These notations mean the state of signal transition from G_k^{i-1} to G_k^i. In detail, “0” means the bus line *i* transmits from signal 0 to signal 0; “1” means the transition from signal 1 to signal 1; “r” means transition from signal 0 to signal 1; “f” means transition from signal 1 to signal 0. For example, “ffrf” means G_k^{i-1} is “1101” and G_k^{i-1} is “0010”.

3.4.2 Generation of RLC table

Before generating the RLC table, we have to extract the parameters, that is, resistance, self capacitance, coupling capacitance, self inductance, and coupling inductance of the signal lines to construct SPICE netlist. By using FastHenry [30] and FastCap [31] with $0.18\,\mu m$ process, we can extract these parameters easily. According to figure 2-5, we construct the bus model and use SPICE to simulate a $1000\,\mu m$-long bus, with $0.24\,\mu m$ width and $0.6\,\mu m$ high. Each $200\,\mu m$-long wire is partitioned as an RLC-segment, where the mutual inductance and capacitance between each two wires are created. The width of each group in RLC table, which is $\frac{n}{m}$ bits.

3.4.3 Compression of RLC table

Obviously, the ROM size is proportional to the width of the RLC table and grows exponentially. So, we have to reduce the ROM size to achieve the smaller area and power consumption. We proposed three ways to reduce the ROM size:

1. Compression of signal state

In Figure 3-4, the first column in the RLC table is the transition type which has four states, that is, *r, f, 1,* and *0*. The state of “r” means the signal line transmits from signal 0 to signal 1; the state of “f” means the signal line transmits from signal 1 to signal 0; the state of “1” means the signal line sticks to signal 1; the state of “0” means the signal line sticks to signal 0. According to experimental result, we found that the state of “0” and “1” does not affect the power consumption, for example, the power consumption of the transition type “ff01” is the same as “ff00”, “ff10”, and “ff11”. For the reason given above, we can prune away the unnecessary signal states to reduce the size of RLC table.

The signal states is compressed from “r”, “f”, “1”, “0” to “r”, “f”, “0”. Suppose the width of RLC table is *k*, and the reduction rate of the ROM size is the following:

$$\frac{4^k - 3^k}{4^k} \qquad (3.4.3.1)$$

For example, if *k* is equal to 8, the number of records in RLC table reduces from 64KByte to 6KByte, and the compression rate is up to 90.625%.

2. Normalization of weight

I use HSPICE as the tool to measure the power consumption, and the output power value is floating point, as shown in Table 3-1. When constructing the RLC table, these simulation data has to be converted to integer to fit the syntax of Verilog. Using more bits to represent the weight results in better improvement of the encoder, but bigger ROM size, and power consumption. Hence, it is important to choose the proper number of bits to store the weight so that we can preserve the accuracy of the simulation, and keep the ROM size and power consumption under control. We use 6 bits to represent the weight to quantize the second column. The normalized RLC Table is shown in Table 3-2.

Table 3-2: Normalized RLC Table

Bus Transition Type	**Weight (Unit: nW)**
ffrf	45
01ff	27
11ff	27
…	…
rrrr	7

3. “Cut” the table into division

In section 3-2, it had been addressed how we choose the partition size. To look after both sides of the number of decision lines and complexity of DBE-RLCT encoder, we choose the size of 8 as the partition width. But we “cut” the RLC table into two identical and smaller RLC tables, which is the size of 4. Besides, we need one extra adder to sum up the weights fetched from both smaller RLC tables, and decide the decision bit. The architecture is shown in Figure 3-4. “InstrNow” means the instruction or data sent in present cycle; “InstrPre” means the instruction or data sent in previous cycle; “InstrNowBar” means the inversion of the instruction or data sent in present cycle. Suppose the original width of RLC table is *S*, and the cut width is $\frac{S}{2}$, the reduction rate of the ROM size is

$$\frac{3^S - 2*3^{\frac{S}{2}}}{3^S} \qquad (3.4.3.2)$$

For example, if the width of RLC table is 8, then the ROM size reduces from 3^8 to $2*3^4$.

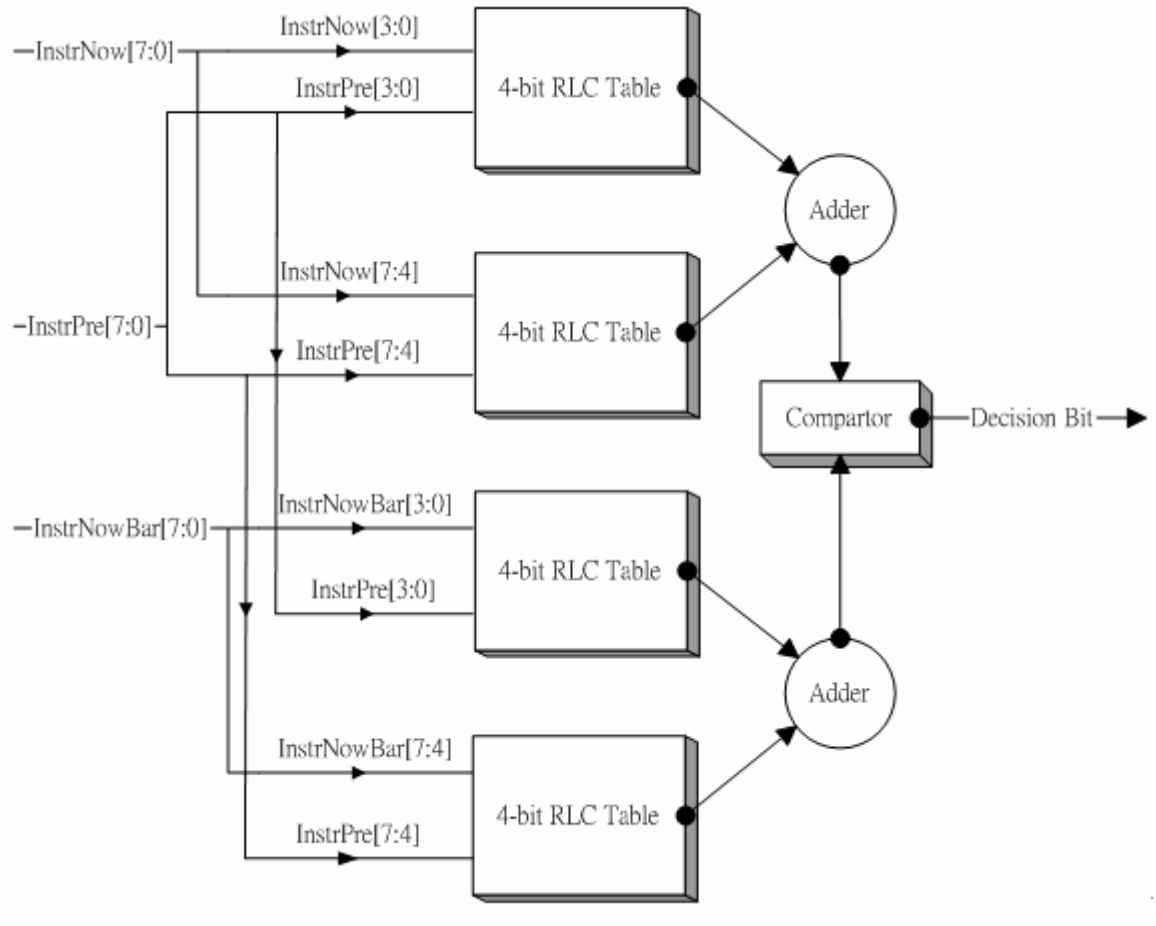


Figure 3-4: Cut the instruction into two partitions, and sum up the weight

Chapter 4：Experimental Setup and Result

4.1 Experimental Environment

In order to show the effectiveness of our method, we have already implemented the encoder and decoder of DBE-RLCT with C++ on Sun workstation. The first step is to construct the RLC table. The inductance and capacitance model is generated by FastHenry [30] and FastCap [31] with 0.18 μm process. We use SPICE to simulate a 1000 μm-long bus, with 0.9 μm width, 1.5 μm high and 0.28 μm spacing which is the minimum value on METAL 3. Each 200 μm-long wire is partitioned as an RLC-segment, where the mutual inductance and capacitance between each two wires are created. The width of each group in RLC table, which is the ($\frac{n}{m}$) bits in definition 1, was set to 8 bits. It can be modified to satisfy the design constraint. With large width, it brings about much power consumption, delay time, and area overhead in the codec design, but it offers better locality and accuracy in bus data transmission. The full bus model was constructed in the same way, but is simplied in some components to shorten the simulation time. We also implemented the encoder and decoder on Sun workstation with 0.18 μm process in Verilog language, synthesize the circuit, and evaluate the codec overhead, which is shown in Table 4-2.

4.2 Experimental Result

Table 4-1 presents the bus power consumption of non-encoded, bus-invert, segmented bus-invert, and DBE-RLCT method. The 100,000 random sequences are used as the input data. Table 4-1 shows that the DBE-RLCT method provides the higher reduction rate of power consumption than other methods, and the average power saving is 7.33%. Figure 4-1 shows the total power consumption of four methods in various sizes of bus width. The result shows that our method is better than the non-encoded, bus-invert, and segmented bus-invert method. Figure 4-2 shows the poser reduction percentage in different methods. These two figures indicate that the proposed method can achieve almost the same crosstalk reduction rate for different sizes of bus width.

From the results, we conclude that the DBE-RLCT method is very effective in

reducing crosstalk effect up to 7.33% reduction of power. This method can be applied to various bus width and bus length, and is suitable for current SoC bus design. The encoder and decoder circuits can be easily designed by simply using some logical gates, adders, and ROMs. The DBE-RLCT method is useful in the real-world design since it uses the realistic RLC table.

Table 4-1: Power consumption (Watt) and saving (%) of non-encoded, bus-invert, segmented bus-invert, and DBE-RLCT (including overhead of the Codec) method in different bus width

Bus Width	Non-Encoded	Bus-Invert		Segmented Bus-Invert		DBE-RLCT	
	Power	Power	Reduction	Power	Reduction	Power	Reduction
8	1340.5	1315.7	1.85	1315.7	1.851	1246	7.051
16	2842.65	2823.5	0.674	2783.2	2.092	2625.88	7.626
24	4341.78	4308.6	0.765	4264.9	1.771	3998.75	7.9
32	5810.56	5808.5	0.036	5707.7	1.771	5374.72	7.5
64	11622.11	11583.7	0.3305	11575	0.405	10858.9	6.567
Avg. Reduction Percentage			0.7312		1.578		7.329

Table 4-2: Codec overhead.

Bus Width	Area (μm^2)	Power (mW)	delay (ns)
8	1856	3.199	2.68
16	3712	6.938	---
24	5568	9.597	---
32	7424	12.796	---
64	14848	25.592	---

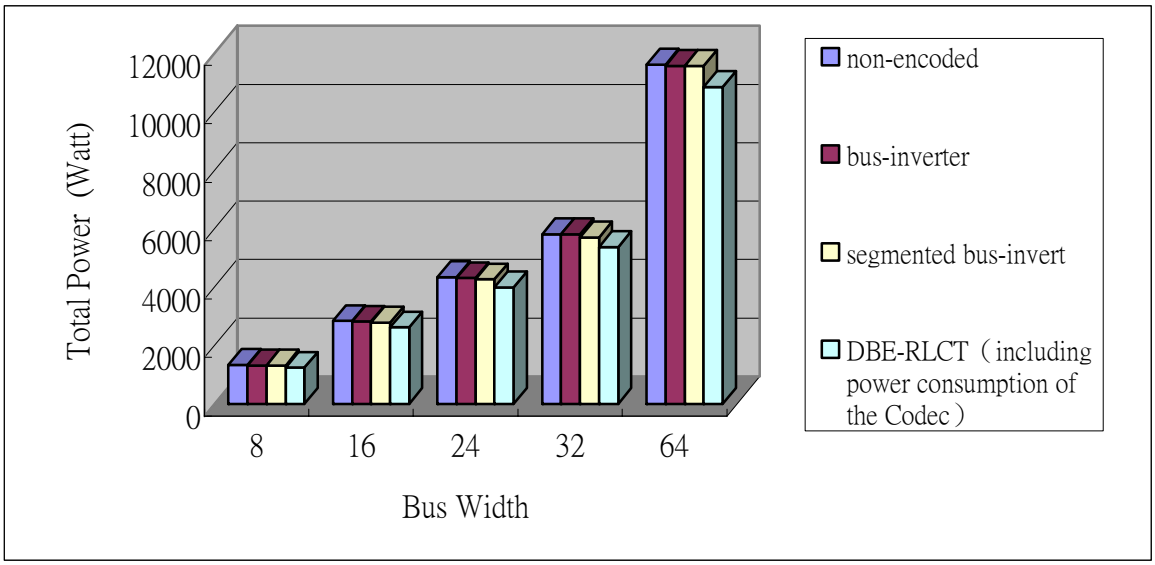


Figure 4-1: Total power consumption of four methods in different bus width

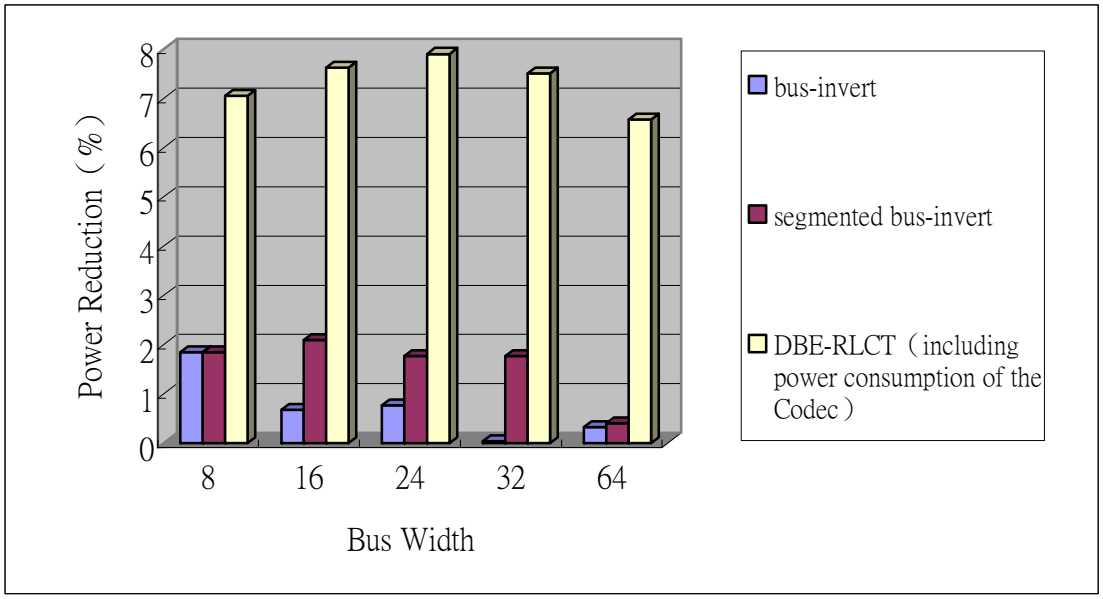


Figure 4-2: Power reduction percentage of four methods in different bus width

$$\text{Power Reduction}\,(=100\times\frac{Power_{org}-Power_{enc}}{Power_{org}})$$

Chapter 5：Conclusion

The DBE-RLCT method based on the realistic RLC table is proposed. When data is prepared to transmit on the bus, we compare it with the previous output data and invert partial input data bits according to the value of RLC table so that we can obtain lower crosstalk effect and power consumption on bus. The experimental result proves that our method is feasible. What should be noticed is that the DBE-RLCT architecture is very flexible since it can be used on arbitrary bus width and the value of the table can be adjusted for different situations. It indicated that DBE-RLCT method is effective for high performance and low power transmission especially for the deep sub-micron bus design, and can be used on NoC (Network On-Chip) in the future.

Bibliography

[1] G. Moore, Progress in Digital Integrated Electronics, IEDM, 1975.

[2] *International Technology Roadmap For Semiconductors, 2001 Edition ; Interconnect*, http://public.itrs.net/Files/2001ITRS/Home.htm.

[3] H. Bakoglu, *Circuits, interconnections, and packaging for VLSI*, Addison-Wesley Pub. Co. Reading, Mass. 1990.

[4] A. Deutsch, et al, ”High-Speed Signal Propagation on Lousy Transmission Lines,” IBM J. Res. Develop., vol. 34, pp. 601-615, 1990

[5] C. Cheng, J. Lilis, S. Lin N.Chang, *Interconnect Analysis and Synthesis*, Wiley Inter-Science 2000.

[6] T. Sakurai, “Design Challenges for 0.1 μm and Beyond”, *Asia and South Pacific Design Automation Conference 2000*, pp. 553-558.

[7] J. Rabaey, *Digital Integrated circuits*, Prentice Hall 1996.

[8] S. Hall, G. Hall, J. McCall, *High-Speed Digital System Design*, Willey Inter-science Publication, John Willey and Sons, 2000.

[9] S. Borkar, “Low power design challenges for the decade”, Asia and South Pacific Design Automation Conference 2001, pp. 293-296.

[10] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, “A coding framework for low-power address and data busses,” *IEEE Trans. On VLSI Systems*, v.7 n.2, p. 212-221, June 1999.

[11] L. Benini, G. D. Micheli, E. Macii, M. Poncino, and S. Quer, “Power optimization of core-based systems by address bus encoding,” *IEEE Trans. on VLSI System*, 6:551-562, Volume: 6, Issue: 4, Dec. 1998, Pages: 554 - 562.

[12] M. Madhu, V. S. Murty, and V. Kamakoti, “Dynamic coding technique for low-power data bus, In *IEEE ISVLSI’03*, pages 252-253, Feb 2003.

[13] Y. Shin, S. Chas, and K. Choi, “Partial bus-invert coding for power optimization of application-specific systems,” *IEEE Trans. on VLSI System*, Volume: 9, Issue: 2, April 2001 Pages: 377 - 383.

[14] E. Musoll, T. Kang, and J. Cortadella, “Working-zone encoding for reducing the energy in microprocessor address buses,” *IEEE Trans. on VLSI System*, 6:568-572, Volume: 6, Issue: 4, Dec. 1998 Pages: 568 – 572.

[15] H. Zhou and D. Wong, “Global routing with crosstalk constraints,” In *in Proceedings of DAC-1998*, June 1998, Pages: 374 – 377.

[16] I. H-R. Jiang, Y.-W. Chang, and J. Y. Jou, “Crosstalk-driven interconnect optimization by simultaneous gate and wire sizing,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, 19:999-1010, Volume: 19 , Issue: 9 , Sept. 2000 Pages:999 - 1010.

[17] E. Naroska, S. J. Ruan, F. Lai, U. Schwiegelshohn, and L. C. Liu, “On optimizing power and crosstalk for bus coupling capacitance using genetic algorithm” In *in Proceedings of ISCAS03*, Volume: 5 , 25-28 May 2003 Pages:V-277 - V-280 vol.5.

[18] A. Deutsch, G. Kopcsay, P. Coteus, C. Surovic, B. Rubin, R. Dunne, T. Gallo, K. Jenkins, L. Terman, R. Dennard, G. Sai-Halasz, B. Krauter, and D. Knebel, “When are trans mission-line effects important for on-chip interconnections?” *IEEE Trans. on Microwave Theory Tech*, 45:1836-3846, Volume: 45, Issue: 10, Oct. 1997, Pages: 1836 - 1846.

[19] K. Lepak, I. Luwandi, and L.He, “Simultaneous shield insertion and net ordering under explicit rlc noise constraint,” In *in Porc. Design Automation Conference*, 2001, pages 199-202, 2001.

[20] D. Bailey and B. Benschneider, “Clocking design and analysis for 600-mhz

alpha microprocessor," *IEEE J. Solid-State Circuit*, Volume: 33, Issue: 11, Nov. 1998, Pages: 1627 - 1633.

[21] L. Vakanas, S. Hasan, A. Cangellaris, and J. L. Prince, "Effects of floating planes in three-dimensional packaging structures on simultaneous switching noise," *IEEE Trans. on Components, Packaging, and Manufacturing Technology*, 21:434-440, Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on [see also Components, Hybrids, and Manufacturing Technology, IEEE Transactions on], Volume: 21, Issue: 4, Nov. 1998, Pages: 434 - 440.

[22] J. Silberman, et al, "A 1.0GHz Single-Issue 64b PowerPC Integer Processor," *in Proc. ISSCC-98, San Francisco*, Feb. 5-7, 1998, pp. 230-231.

[23] E. E. Davidson, B. D. McCredie, and W. V. Vilkelis, "Long Lossy Lines(L^3) and Their Impact Upon Chip Performance," *IEEE Trans. Comp. Packaging, Manuf. Technol.-Part B*, Vol. 20, No .4, pp.361-375, Nov. 1997.

[24] C. K Cheng, J. Lillis, S. Lin, and N. Chang, "Interconnect Analysis and Synthesis," *John Wiley & Sons, Inc.*, 1999.

[25] J. Torres, "Advanced Copper Interconnections for Silicon COMS Technologies," Applied Surface Science, Vol. 91, pp.112-123, Oct. 1995.

[26] A. Deutsch, et all, "When are Transmission-Line Effects Important for On-Chip Wiring," IEEE Trans. Microwave Theory Tech., vol. 45, pp. 1836-1846, Oct. 1997.

[27] A. Deutsch, et all, "The Importance of Inductance and Inductive Coupling for On-Chip Wiring," Proc. IEEE 6-th Electrical Performance of Electronic Packaging, pp. 53-56, Oct. 1997.

[28] Rung-Bin Lin; Chi-Ming Tsai, **"**Theoretical analysis of bus-invert coding" *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, Volume: 10, Issue: 6, Dec. 2002, Pages: 929 – 934.

[29] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power i/o, "*IEEE Trans. On VLSI System*, 3:49-58, Volume: 3, Issue: 1, March 1995, Pages: 49 - 58.

[30] M. Kamon, and et al, “Fasthenry: a multiple-acceler-ated 3-d inductance extraction program,” *IEEE Trans on MTT,* Volume: 42, Issue: 9, Sept. 1994, Pages: 1750 – 1758.

[31] K. Nabors and J. White, “Fastcap: a multiple-acceler-ated 3-d capacitance extraction program,” *IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems,* Volume: 10, Issue: 11, Nov. 1991, Pages: 1447 - 1459.

[32] Frederick W. Grover, *Inductance Calculations Working Formulas and Tables*, Dover Publications, 1946.

[33] Dr. Eric Bogatin, *What Really Is Inductance?,* Bogatin Enterprises Oct 30, 1999.

[34] Kaushik Gala, David Blaauw, Junfeng Wang, Vladimir Zolotov, Min Zhao, “Inductance 101: Analysis and Design Issues”, Design Automation Conference, 2001

[35] Ruehli, A. E., “Inductance Calculations in a Complex Integrated Circuit Environment,” IBM Journal of Research and Development, Sept. 1972, pp 470-481

[36] Gala, K., et al., “On Chip Inductance Modeling and Analysis.” DAC, June 2000, pp63-68