THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Data Encoding Technique Using Gray Code in Network-on-Chip

S. Kavitha
Student, PG Scholar/VLSI Design, Karpagam University, Coimbatore, India

Abstract:
As technology varies the power dissipated by the links of a network-on-chip (NOC) due to noise increases the risk of errors in the communication subsystem. In this paper, we present a data encoding technique using gray code input aimed at reducing the errors and power dissipation by the links of an NOC. The proposed scheme uses the binary to gray conversion at the transmitter and gray to binary conversion at the receiver. An experimental result from both synthetic and real traffic scenarios shows the effectiveness of the proposed schemes which allows to save the power up to 21.6% and reduces the noise or error during the transmission through the encoder using Gray code.

Key words: *Binary to Gray conversion, data encoding, interconnection on chip, low power, network-on-chip (NoC), power analysis*

1. Introduction

SPEED and power consumption of on-chip interconnect network have become important in advanced CMOS technologies. The advances in fabrication technology allow designers to implement a whole system on a single chip, but the inherent design complexity of such systems makes it hard to fully explore the technology potential. Thus, the design of Systems-on-Chip (SoCs) is usually based on the reuse of predesigned and pre-verified intellectual property core that are interconnected through special communication resources that must handle very tight performance and area constraints. In addition to those application-related constraints, deep submicron effects pose physical design challenges for long wires and global on-chip communication. The growing market for portable battery-powered devices adds a new dimension, power, to the VLSI design space, previously characterized by speed and area. The power consumption is related to battery life as well as costly package and heat sink requirements for high-end devices.

In order to ensure the final system complies to the desired function, thermal and cost requirements, the power consumption issues must be addressed during the design of all subsystems in a SoC, including the interconnect structure. One problem related to power consumption in busses is the capacitances induced by long wires. Such problem is minimized in NoCs, since point-to-point short wires are used between routers. However, NoCs consumes power in routers, diminishing the apparent advantage in terms of power when compared to busses. The power consumption in a NoC grows linearly with the amount of bit transitions in subsequent data packets sent through the interconnect architecture. One way to reduce power consumption in NoCs, in both wires and logic, is to reduce the switching activity by means of coding schemes.

Network-on-Chip (NoC) is generally viewed as the ultimate solution for the design of modular and scalable communication architectures. A NoC-based architecture for communication promises flexibility in network topology, the support of advanced flow-control and switching techniques, routing algorithms, and the possibility of guarantying quality-of-service requirements. For the bus-based architectures these advantages comes at the cost of increase in complexity which pushes the communication system to become one of the main elements of a SoC which strongly impact the cost, power, and performance figures of the overall system.

As the number of transistors that can be placed on an IC is increasing exponentially, i.e., doubling approximately every two years. In 1965, Gordon Moore states that “The complexity for minimum component costs has increased at a rate of roughly a factor of two per year”.

Data encoding is often employed to decrease the number of bit transitions over interconnects. Popular methods include Bus Invert (BI), adaptive coding, gray coding and transition method.

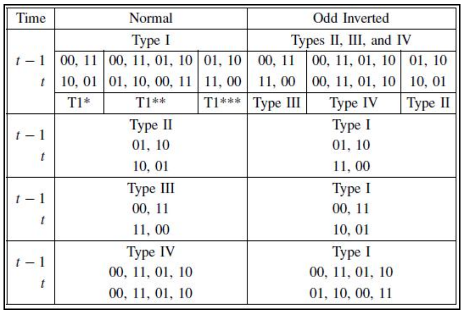
In this paper, we focus on technique aimed at reducing the power dissipated by the links and error during transmission of the signals. In particular we present a data encoding technique using gray code input. The result shows that by using the proposed encoding scheme up to 21.6% of power can be saved.

The rest of this paper is organized as follows. We briefly discuss related works in Section II, while Section III presents an overview of the proposed data encoding schemes. The proposed data encoding schemes along with possible hardware implementations and their analysis are described in Section IV. In Section V, the results for the hardware overhead, power and energy savings, and performance reduction of the proposed data encoding schemes are compared with those of other approaches. Finally, this paper is concluded in Section VI.

2. Related Works and Contributions

In the next several years, the availability of chips with 1000 cores is foreseen [6]. In these chips, a significant fraction of the total system power budget is dissipated by interconnection networks. Therefore, the design of power-efficient interconnection networks has been the focus of many works published in the literature dealing with NoC architectures. These works concentrate on different components of the interconnection networks such as routers, NIs, and links. Since the focus of this paper is on reducing the power dissipated by the links, in this section, we briefly review some of the works in the area of link power reduction. These include the techniques that make use of shielding [7], [8], increasing line-to-line spacing [9], [10], and repeater insertion [11]. They all increase the chip area. The data encoding scheme is another method that was employed to reduce the link power dissipation. The data encoding techniques may be classified into two categories. In the first category, encoding techniques concentrate on lowering the power due to self-switching activity of individual bus lines while ignoring the power dissipation owing to their coupling switching activity. In this category, bus invert (BI) [12] and INC-XOR [13] have been proposed for the case that random data patterns are transmitted via these lines. On the other hand, gray code [14], T0 [15], working-zone encoding [16], and T0-XOR [17] were suggested for the case of correlated data patterns. Application-specific approaches have also been proposed [18]–[22].

This category of encoding is not suitable to be applied in the deep sub-micron meter technology nodes where the coupling capacitance constitutes a major part of the total interconnect capacitance. This causes the power consumption due to the coupling switching activity to become a large fraction of the total link power consumption, making the fore mentioned techniques, which ignore such contributions, inefficient [23]. The works in the second category concentrate on reducing power dissipation through the reduction of the coupling switching [10], [22]–[29]. Among these schemes [10], [24]– [28], the switching activity is reduced using many extra control lines. For example, the data bus width grows from 32 to 55 in [24]. The techniques proposed in [28] and [29] have a smaller number of control lines but the complexity of their decoding logic is high. The technique described in [29] is as follows: first, the data are both odd inverted and even inverted, and then transmission is performed using the kind of inversion which reduces more the switching activity. In [29], the coupling switching activity is reduced up to 39%. In this paper, compared to [29], we use a simpler decoder while achieving a higher activity reduction.



Time	Normal			Odd Inverted		
	Type I			Types II, III, and IV		
$t - 1$ t	00, 11 10, 01	00, 11, 01, 10 01, 10, 00, 11	01, 10 11, 00	00, 11 11, 00	00, 11, 01, 10 00, 11, 01, 10	01, 10 10, 01
	T1*	T1**	T1***	Type III	Type IV	Type II
$t - 1$ t	Type II 01, 10 10, 01			Type I 01, 10 11, 00		
$t - 1$ t	Type III 00, 11 11, 00			Type I 00, 11 10, 01		
$t - 1$ t	Type IV 00, 11, 01, 10 00, 11, 01, 10			Type I 00, 11, 01, 10 01, 10, 00, 11		

Table 1: Effect Of Odd Inversion On Change Of Transition Types

Let us now discuss in more detail the works with which we compare our proposed schemes. In [12], the number of transitions from 0 to 1 for two consecutive flits (the flit that just traversed and the one which is about to traverse the link) is counted. If the number is larger than half of the link width, the inversion will be performed to reduce the number of 0 to 1 transitions when the flit is transferred via the link. This technique is only concerned about the self-switching without worrying the coupling switching. Note that the coupling capacitance in the state-of–the-art silicon technology is considerably larger (e.g., four times) compared with the self-capacitance, and hence, should be considered in any scheme proposed for the link power reduction.

In addition, the scheme was based on the hop-by-hop technique, and therefore, encoding/ decoding is performed in each node.

The scheme presented in [26] dealt with reducing the coupling switching. In this method, a complex encoder counts the number of Type I (Table I) transitions with a weighting coefficient of one and the number of Type II transitions with the weighting coefficient of two. If the number is larger than half of the link width, the inversion will be performed. In addition to the complex encoder, the technique only works on the patterns whose full inversion leads to the link power reduction while not considering the patterns whose full inversions may lead to higher link power consumption. Therefore, the link power reduction achieved through this technique is not as large as it could be. This scheme was also based on the hop-by-hop technique.

In another coding technique presented in [25], bunches of four bits are encoded with five bits. The encoded bits were isolated using shielding wires such that the occurrence of the patterns "101" and "010" were prevented. This way, no simultaneous Type II transitions in two adjacent pair bits are induced. This technique effectively reduces the coupling switching activity. Although the technique reduces the power consumption considerably, it increases the data transfer time, and, hence, the link energy consumption. This is due to the fact that for each four bits, six bits are transmitted which increases the communication traffic. This technique was also based on the hop-by-hop approach.

A coding technique that reduces the coupling switching activity by taking the advantage of end-to-end encoding for wormhole switching has been presented in [23]. It is based on lowering the coupling switching activity by eliminating only Type II transitions. In this paper, we present data encoding schemes, which focused on reducing the errors during the transition from transmitter to receiver which leads to the higher power saving.

3. Overview of the Proposal

The basic idea of the proposed approach is encoding the flits before they are injected into the network with the goal of minimizing the self-switching activity and the coupling switching activity in the links traversed by the flits. In fact, self-switching activity and coupling switching activity are responsible for link power dissipation. In this paper, we refer to the end-to-end scheme. This end-to-end encoding technique takes advantage of the pipeline nature of the wormhole switching technique [4]. Note that since the same sequence of flits passes through all the links of the routing path, the encoding decision taken at the NI may provide the same power saving for all the links. For the proposed scheme, an encoder and a decoder block are added to the NI. Except for the header flit, the encoder encodes the outgoing flits of the packet such that the power dissipated by the inter-router point-to-point link is minimized [23].

4. Proposed Encoding Schemes

In this section, we present the proposed encoding scheme whose goal is to reduce power dissipation by minimizing the coupling transition activities on the links of the interconnection network and to reduce errors using gray code. Fig.1 shows the block diagram of the proposed encoding scheme. It consists of a binary to gray converter, encoder, decoder and gray to binary converter.

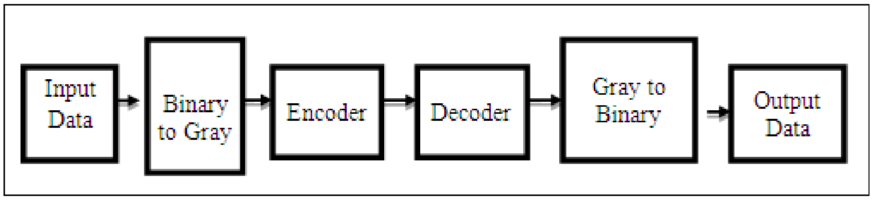


Figure 1: Block Diagram of the proposed encoding scheme

The binary to gray converter is used to convert the input data to gray number since it is used to facilitate error correction in digital communication.

4.1. Binary to Gray Converter

The proposed Binary to Gray converter is used to convert the binary data to gray data. The reflected binary code, also known as Gray code is a binary numeral system where only one bit differs between two successive values. The reflected binary code was mainly designed to prevent spurious output from electromechanical switches. Gray codes are widely used to facilitate error correction in digital communications such as digital terrestrial television and some cable TV systems.

4.1.1. Problem with natural binary code

The problem with natural binary codes is that, with real (mechanical) switches, the switches will change states exactly in synchrony. During the transition between the two states 011 - 100, all three switches change state. When all the states are changing some spurious position will be read with the switches. The transition might look like 011 — 001 — 101 — 100 even without key bounce. When the switches appear to be in position 001, it cannot identify that the "real" position is 001, or between two other positions of the transitional state. If the output feeds into a sequential system (possibly via combinational logic) then the sequential system may store a false value. The reflected binary code solves spurious problem by changing only one switch at a time, so that any ambiguity of position will never occur.

DECIMAL	BINARY	GRAY
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Table 2: 4-Bit Binary to gray converter

4.1.2. Conversion from binary to gray

Gray code is a non-weighted code which belongs to a class of codes called minimum change codes. From this table we can obtain the equivalent gray code of the decimal values. These are the steps which will make you understand how the codes are formed.

- In case of gray code one bit will change from its previous in each step. One thing must be kept in mind that the change of bit always occurs from the right side i.e., from L.S.B towards the M.S.B. For the decimal value 1 the equivalent binary code is 0001 when it is converted to gray code the first three bits are constant i,e., 000 and the fourth bit changes from 0 to 1. We know that for binary digit possible combination is 0 and 1, so keeping first three bit constant the possible combination of 4th bit is over for decimal 0 and 1 respectively.
- Now move to the next bit from L.S.B i.e 3rd bit, that change from 0 to 1 which is the decimal equivalent for 2. Now one more combination is left for the fourth bit keeping the first three digits as constant i.e 001. We can change 4th bit from 1 to 0. Thus the gray code for decimal number 3 is 0010.
- Traverse to the next code. Here we can do only one thing i.e., we can change the second bit as all possible combinations are over. But changing third bit would give the equivalent gray code 0000 which has occurred earlier. So you must remember that a number occurring previously must not be repeated. So the equivalent code for 4 will be 0110. Here only the second bit has changed from the previous code. Now again we will keep first and second bit constant and find the possible combinations of the third and the fourth bit by only changing 1 bit in each steps.

Binary to gray code conversion is a very simple process. There are several steps to do these types of conversions. Steps given below elaborate on the idea on this type of conversion.

- The M.S.B. of the gray code will be exactly equal to the first bit of the given binary number.
- Now the second bit of the code will be exclusive-or of the first and second bit of the given binary number, i.e if both the bits are same the result will be 0 and if they are different the result will be 1.

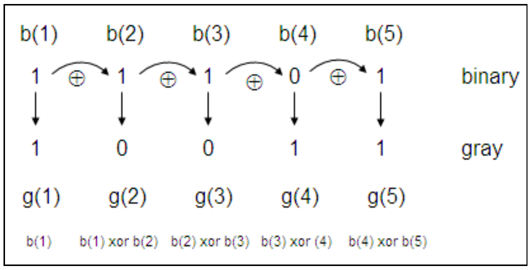


Figure 2: Binary to Gray Conversion

- The third bit of gray code will be equal to the exclusive-or of the second and third bit of the given binary number. Thus the Binary to gray code conversion goes on. One example given below can make your idea clear on this type of conversion.

The most significant bit (MSB) in Gray is taken directly from the MSB in binary. The rest of the Gray bits comes from a XOR operation between the precedent binary bit (b(i-1)) and the current binary bit (b(i)). In the case shown in the figure above:
G(3) = B(3)
G(2) = B(2) XOR B(3)
G(1) = B(1) XOR B(2)
G(0) = B(0) XOR B(1)

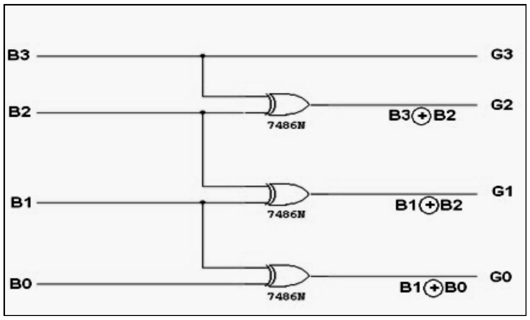


Figure 3: Logic diagram of Binary to Gray

4.2. Digital Encoding & Decoding Scheme

An encoder is a device, circuit, transducer, software program, algorithm or person that converts information from one format or code to another, for the purposes of standardization, speed, secrecy, security or compressions. A simple encoder circuit can receive a single active input out of 2n input lines generate a binary code on n parallel output lines. It is one of the basic elements of any digital communication system

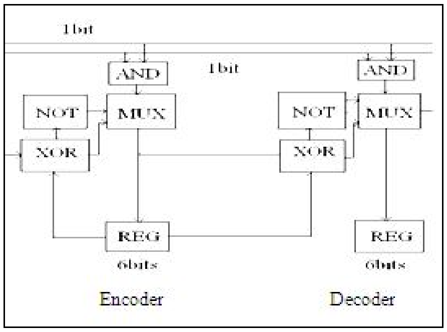


Figure 4: Digital Encoding & Decoding Scheme

The information sequence is passed through the channel encoder. The purpose of the channel encoder is to introduced, in controlled manner, some redundancy in the binary information sequence that can be used at the receiver to overcome the effects of noise and interference encountered in the transmission on the signal through the channel.

A decoder is a device which does the reverse operation of an encoder, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode. It is a combinational circuit that converts binary information from n input lines to a maximum of 2n unique output lines. The sequence of numbers passes through the channel decoder in communication system which attempts to reconstruct the original information sequence from the knowledge of the code used by the channel encoder and the redundancy contained in the received data. The encoder and decoder are responsible for implementing the tolerance against transient faults.

4.3. Gray to Binary Converter

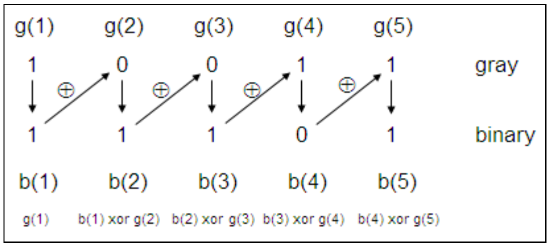


Figure 5: Gray to Binary conversion

B(0) = G(0)
B(1) = B(0) XOR G(1)
B(2) = B(1) XOR G(2)
B(3) = B(2) XOR G(3)
B(4) = B(3) XOR G(4)

Gray code to binary conversion is again very simple and easy process. Following steps can make your idea clear on this type of conversions.

- The M.S.B of the binary number will be equal to the M.S.B of the given gray code.
- Now if the second gray bit is 0 the second binary bit will be same as the previous or the first bit. If the gray bit is 1 the second binary bit will alter. If it was 1 it will be 0 and if it was 0 it will be 1.
- This step is continued for all the bits to do Gray code to binary conversion.

The M.S.B of the binary will be 0 as the M.S.B of gray is 0. Now move to the next gray bit. As it is 1 the previous binary bit will alter i.e it will be 1, thus the second binary bit will be 1. The third bit of the gray code is again 1 thus the previous bit i.e the second binary bit will again alter and the third bit of the binary number will be 0. Now, 4th bit of the given gray is 0 so the previous binary bit will be unchanged, i.e 4th binary bit will be 0. Now again the 5th grey bit is 1 thus the previous binary bit will alter, it will be 1 from 0.

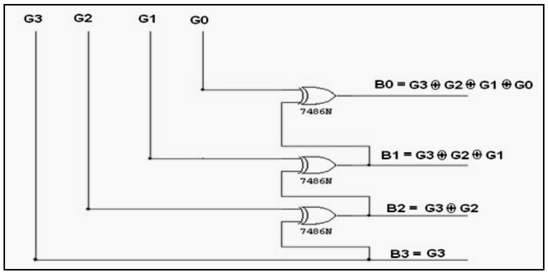


Figure 6: Logic diagram of Gray to Binary

5. Results and Discussion

5.1. Conversion Analysis

The binary and gray conversions were designed in Verilog HDL described at the RTL level, synthesized with synopsis design compiler and mapped onto an UMC 65-nm technology library. In our study, the area and power of the proposed encoder is compared against existing encoder scheme I, scheme II, and scheme III.

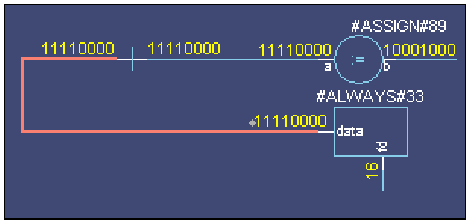
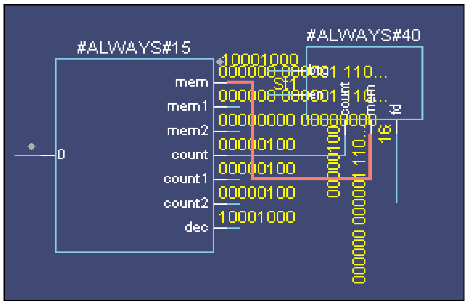


Figure 7: Dataflow diagram of Decoded data

5.2. Data Analysis

The 8 bit Input data of 00001111 is encoded and sent through communication channel and decoded at the receiver without any errors and noise.



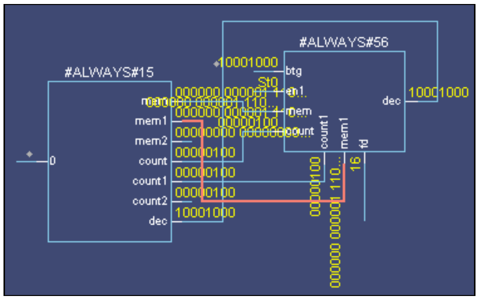


Figure 8: Dataflow diagram of Encoded data Decoder
Figure 9: Dataflow diagram of Decoded data

5.3. Power Versus Performance

The trade-off between the reduction of the average power dissipation of the communication system with the completion time (i.e., the amount of the time needed to drain a given amount of traffic volume) is an important characteristic of the system. The percentage increase of completion time is defined as the percentage increase of the time needed to drain a given amount of traffic. The characteristic for each encoding scheme has been plotted.

S.No	Parameters	Encoding Schemes	Proposed Scheme
1.	Device	5M40Z	5M40Z
2.	Clocks	1.48	1.48
3.	$I_{CCPOWERUP}$	40.0	40.0
4.	I_{CCIO}	0.08	0.04
5.	$P_{STANDBY}$	0.04	0.04
6.	P_{TOTAL}	2.16	2.05

Table 3: Power Versus Performance

6. Conclusion

In this paper, we have presented a set of new data encoding schemes aimed at reducing the power dissipated by the links of a NoC. In fact, links are responsible for a significant fraction of the overall power dissipated by the communication system. In addition, their contribution is expected to increase in future technology nodes. As compared to the previous encoding schemes proposed in the literature, the rationale behind the proposed schemes is to minimize not only the switching activity, but also (and in particular) the coupling switching activity which is mainly responsible for link power dissipation in the deep sub-micronmeter technology regime. The proposed encoding schemes are agnostic with respect to the underlying NoC architecture in the sense that their application does not require any modification neither in the routers nor in the links. An extensive evaluation has been carried out to assess the impact of the encoder and decoder logic in the NI. The encoders implementing the proposed schemes have been assessed in terms of power dissipation and silicon area. The impacts on the performance, power, and energy metrics have been studied using a cycle- and bit accurate NoC simulator under both synthetic and real traffic scenarios. Overall, the application of the proposed encoding schemes allows savings up to 21.6% of power dissipation and reduces the error during transmission without any significant performance degradation.

7. References

1. International Technology Roadmap for Semiconductors. (2011) [Online]. Available: http://www.itrs.net
2. M. S. Rahaman and M. H. Chowdhury, “Crosstalk avoidance and errorcorrection coding for coupled RLC interconnects,” in Proc. IEEE Int. Symp. Circuits Syst., May 2009, pp. 141–144.
3. W. Wolf, A. A. Jerraya, and G. Martin, “Multiprocessor system-on-chip MPSoC technology,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 27, no. 10, pp. 1701–1713, Oct. 2008.
4. L. Benini and G. De Micheli, “Networks on chips: A new SoC paradigm,” Computer, vol. 35, no. 1, pp. 70–78, Jan. 2002.
5. S. E. Lee and N. Bagherzadeh, “A variable frequency link for a power aware network-on-chip (NoC),” Integr. VLSI J., vol. 42, no. 4, pp. 479–485, Sep. 2009.
6. Vittal and M. Marek-Sadowska, “Crosstalk reduction for VLSI,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 16, no. 3, pp. 290–298, Mar. 1997.
7. M. Ghoneima, Y. I. Ismail, M. M. Khellah, J. W. Tschanz, and V. De, “Formal derivation of optimal active shielding for low-power on-chip buses,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 25, no. 5, pp. 821–836, May 2006.
8. L. Macchiarulo, E. Macii, and M. Poncino, “Wire placement for crosstalk energy minimization in address buses,” in Proc. Design Autom. Test Eur. Conf. Exhibit., Mar. 2002, pp. 158–162.
9. R. Ayoub and A. Orailoglu, “A unified transformational approach for reductions in fault vulnerability, power, and crosstalk noise and delay on processor buses,” in Proc. Design Autom. Conf. Asia South Pacific, vol. 2. Jan. 2005, pp. 729–734.
10. K. Banerjee and A. Mehrotra, “A power-optimal repeater insertion methodology for global interconnects in nanometer designs,” IEEE Trans. Electron Devices, vol. 49, no. 11, pp. 2001–2007, Nov. 2002.
11. M. R. Stan and W. P. Burleson, “Bus-invert coding for low-power I/O,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 3, no. 1, pp. 49–58, Mar. 1995.
12. S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, “A coding framework for low-power address and data busses,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 7, no. 2, pp. 212–221, Jun. 1999.
13. L. Su, C. Y. Tsui, and A. M. Despain, “Saving power in the control path of embedded processors,” IEEE Design Test Comput., vol. 11, no. 4, pp. 24–31, Oct.–Dec. 1994.
14. L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, “Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems,” in Proc. 7th Great Lakes Symp. VLSI, Mar. 1997, pp. 77–82.
15. Musoll, T. Lang, and J. Cortadella, “Working-zone encoding for reducing the energy in microprocessor address buses,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 6, no. 4, pp. 568–572, Dec. 1998.
16. W. Fornaciari, M. Polentarutti, D. Sciuto, and C. Silvano, “Power optimization of system-level address buses based on software profiling,” in Proc. 8th Int. Workshop Hardw. Softw. Codesign, May 2000, pp. 29–33.
17. L. Benini, G. De Micheli, E. Macii, M. Poncino, and S. Quer, “Power optimization of core-based systems by address bus encoding,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 6, no. 4, pp. 554–562, Dec. 1998.
18. L. Benini, A. Macii, M. Poncino, and R. Scarsi, “Architectures and synthesis algorithms for power-efficient bus interfaces,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 19, no. 9, pp. 969–980, Sep. 2000.
19. Ascia, V. Catania, M. Palesi, and A. Parlato, “Switching activity reduction in embedded systems: A genetic bus encoding approach,” IEEE Proc. Comput. Digit. Tech., vol. 152, no. 6, pp. 756–764, Nov. 2005.
20. R. Siegmund, C. Kretzschmar, and D. Muller, “Adaptive Partial Businvert encoding for power efficient data transfer over wide system buses,” in Proc. 13th Symp. Integr. Circuits Syst. Design, Sep. 2000, pp. 371–376.
21. S. Youngsoo, C. Soo-Ik, and C. Kiyoung, “Partial bus-invert coding for power optimization of application-specific systems,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 9, no. 2, pp. 377–383, Apr. 2001.
22. M. Palesi, G. Ascia, F. Fazzino, and V. Catania, “Data encoding schemes in networks on chip,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 5, pp. 774–786, May 2011.
23. G. Lyuh and T. Kim, “Low-power bus encoding with crosstalk delay elimination,” IEE Proc. Comput. Digit. Tech., vol. 153, no. 2, pp. 93–100, Mar. 2006.

24. P. P. Pande, H. Zhu, A. Ganguly, and C. Grecu, "Energy reduction through crosstalk avoidance coding in NoC paradigm," in Proc. 9th EUROMICRO Conf. Digit. Syst. Design Archit. Methods Tools, Sep. 2006, pp. 689–695.
25. K. W. Ki, B. Kwang Hyun, N. Shanbhag, C. L. Liu, and K. M. Sung, "Coupling-driven signal encoding scheme for low-power interface design," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, Nov. 2000, pp. 318–321.
26. L. Rung-Bin, "Inter-wire coupling reduction analysis of bus-invert coding," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 7, pp. 1911–1920, Aug. 2008.
27. Z. Khan, T. Arslan, and A. T. Erdogan, "Low power system on chip bus encoding scheme with crosstalk noise reduction capability," IEE Proc. Comput. Digit. Tech., vol. 153, no. 2, pp. 101–108, Mar. 2006.
28. Z. Yan, J. Lach, K. Skadron, and M. R. Stan, "Odd/even bus invert with two-phase transfer for buses with coupling," in Proc. Int. Symp. Low Power Electron. Design, 2002, pp. 80–83.
29. P. Fan and C. H. Fang, "Efficient RC low-power bus encoding methods for crosstalk reduction," Integr. VLSI J., vol. 44, no. 1, pp. 75–86, Jan. 2011.