International Journal of Recent Trends in Engineering & Research
ISSN : 2455-1457 Impact Factor : 3.344

LOW POWER AND HIGH SPEED DATA ENCODING TECHNIQUE IN NoC

Mrs. Gopika. V[1], Ms P. Radhika[2]
[1,2]Assistant Professor, PPGIT, Coimbatore, Tamil Nadu, India

Abstract - Network on Chip is a communication subsystem between IP cores in a System on Chip. On chip networks are currently used for applications like parallel processing, soft computing and so on. NoC is constructed by multiple point to point data links interconnected by switches. The main challenge faced by NoC is the power dissipated during self and coupling switching activity while data bits are transmitted. In this paper we present a data encoding scheme which reduce the power dissipated in the links and transmit data as gray codes for further reduction in power and increase in speed. The data packets are encoded before they enter into the network links so that no modification in routers or network links is needed.

Index Terms - Coupling switching activity, data encoding, interconnection on chip, low power, network-on-chip (NoC), power analysis.

I. INTRODUCTION

An integrated solution to challenging design problems in telecommunication, multimedia and consumer electronics domain is provided by System on Chip. Success of SoC depends upon design and process technologies and ability to interconnect components appropriately. The major challenge faced by designers is to provide functionally correct and reliable interacting components. The performance and energy consumption of the network depends upon physical interconnections.

Network on Chip is the system used for communication between IP cores in a SoC. It consists of routers interconnected by communication channels. The basic elements that form NoC are Network Interfaces, routers and links. With shrinking technology the power dissipated in links is more relevant than power dissipated in routers and network interfaces. As the complexity of the design increases, total length of the interconnecting wires increases resulting in more transmission delay and power consumption. In addition to this the distance between wires reduces which results in an increase in coupling capacitance.

Focus is given to various data encoding schemes as a way to reduce power dissipated by network links. The basic idea is to encode the data before it is injected into the network for reduction in switching activity.

II. OVERVIEW OF THE PROPOSAL

Our proposal exploits the pipeline nature of wormhole switching technique to implement an end-to-end encoding/decoding scheme. In our proposal data are encoded before transmission and are decoded at the destination. This makes the approach transparent with respect to the underlying NoC fabric as it does not require any modification of the router architecture. Encoder and decoder architecture proposed help to reduce both self and coupling switching activity.

Differently from the previous approaches on data encoding in NoCs, in our proposal a binary to gray code converter is implemented to transmit data bits as gray codes. Gray code which is also known as reflected binary code is a binary numeral system where two successive values differ in only one bit. This code was designed to prevent spurious output from electromechanical switches. Transfer of data bits as gray codes reduces the switching activity thereby still reducing the power dissipated in the network links.

III. PROPOSED ENCODING SCHEME

The proposed encoding scheme is presented in this section with a goal to reduce coupling switching activity in network links.

The data is classified into four types based on bit transitions.

Type I transition - occurs when one of the lines switches when the other remains unchanged

Type II transition - occurs when one line switches from low to high while the other makes transition from high to low
Type III transition - corresponds to the case where both lines switch simultaneously
Type IV transition - both lines do not change

Thus by converting Type I, II and III transitions into Type IV transition we will be able to reduce the switching activity. For converting from one transition type to other we have to consider odd, even or full inversion of the bits. These inversions have different effect on each type of transitions.

Table – I
Effect of odd inversion on Type I transition

Normal			Odd Inverted		
Type I			Types II, III, and IV		
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

There are three types of Type I transitions which are named as T1*, T1**, T1***. From Table – I we can come to a conclusion that Type I transitions when odd inverted get converted into Type III, Type IV and Type II transitions. Thus by odd inverting T1* and T1** we can convert them to Type III and Type IV transitions respectively, thus reducing the switching activity. As T1*** transitions are getting converted into Type II transitions, there is no much reduction in switching activity in this case. We have to find an alternate solution to reduce T1*** transitions.

From Table – II it can be observed that when even inverted T1*** transitions are getting converted into Type III transitions thereby reducing switching activity. Thus by performing odd and even inversion on flits we are able to reduce Type I transitions. However we have to find a solution to reduce Type II transitions.

Table – II
Effect of even inversion on Type I transition

Normal			Even Inverted		
Type I			Types II, III, and IV		
01, 10 00, 11	00, 11, 01, 10 10, 01, 11, 00	00, 11 01, 10	01, 10 10, 01	00, 11, 01, 10 00, 11, 01, 10	00, 11 11, 00
T1*	T1**	T1***	Type II	Type IV	Type III

Table – III
Effect of full inversion on Type II transition

Normal	**Full Inverted**
Type II	**Type IV**
01, 10 **10, 01**	**01, 10** **01, 10**

From Table – III it can be observed Type II transitions when full inverted get converted into Type IV transitions. Therefore by full inversion of Type II transitions switching activity can be much reduced. Thus by applying odd, even or full inversion on flits we are able to reduce the switching activity to a great extend. As switching activity reduces power dissipated when data flits are transmitted gets reduced.

The proposed encoder block diagram is shown in the fig. 1. We consider a link width of w bits. If no encoding is used the body flits are grouped in w bits. If no encoding is used, the body flits are grouped in w bits

by the NI and are transmitted via the link. In our approach, one bit of the link is used for the inversion bit, which indicates if the flit traversing the link has been inverted or not. More specifically, the NI packs the body flits in w − 1 bits. The encoding logic E, which is integrated into the NI, is responsible for deciding if the inversion should take place and performing the inversion if needed. To make the decision, the previously encoded flit is compared with the current flit being transmitted.

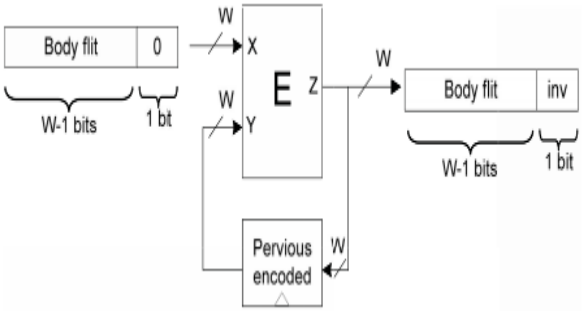


Fig. 1. Block diagram of proposed encoder in transmitter side

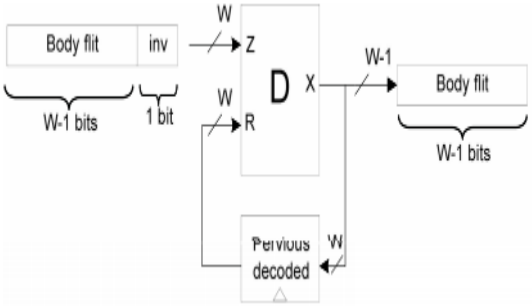


Fig. 2. Block diagram of proposed decoder in receiver side

Proposed encoder architecture:

The w − 1 bits of the incoming (previous encoded) body flit are indicated by Xi (Yi), i = 0, 1, . . . ,w − 2. The w th bit of the previously encoded body flit is indicated by inv which shows if it was inverted (inv = 1) or left as it was (inv = 0). The first stage of the encoder determines the transition types while the second stage is formed by a set of 1s blocks which count the number of ones in their inputs. In the encoding logic, each Ty block takes the two adjacent bits of the input flits (e.g., X1X2Y1Y2, X2X3Y2Y3, X3X4Y3Y4, etc.) and sets its output to “1” if any of the transition types of Ty is detected. This means that the odd inverting for this pair of bits leads to the reduction of the link power dissipation (Table I). The Ty block may be implemented using a simple circuit. Te blocks which determine if any of the transition types of

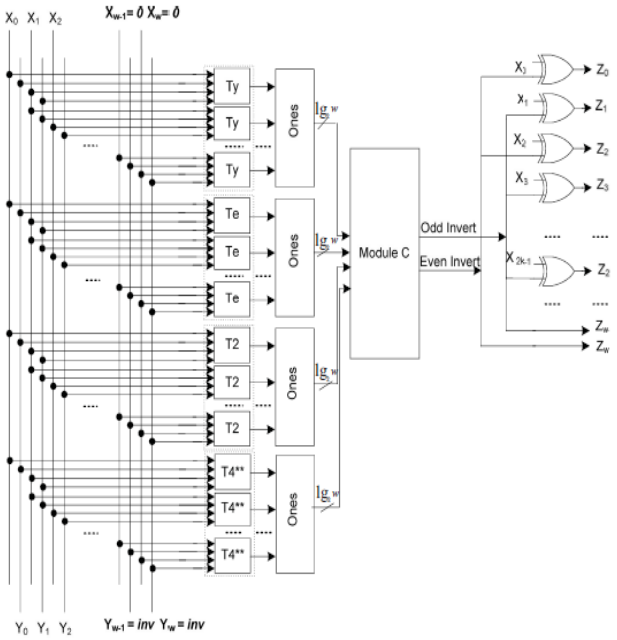


Fig. 3. Encoder architecture

T2, T 1**, and T 1*** is detected for each pair bits of their inputs. For these transition types, the even invert action yields link power reduction. Again, we have four Ones blocks to determine the number of detected transitions for each Ty, Te, T2, T 4, blocks. The output of the Ones blocks are inputs for Module C. This module determines if odd, even, full, or no invert action corresponding to the outputs "10," "01," "11," or "00," respectively, should be performed. In this paper, Module C was designed based on the conditions given in (28), (29), and (30).

Binary to gray code conversion:

The data flits received from the encoder are in binary format. They are converted into gray code using binary to gray code converter and transmitted as gray codes. Advantage of using gray code is that change in only one bit is needed to change the output which will reduce the switching activity.

The problem with natural binary codes is that, with physical, mechanical switches, it is very unlikely that switches will change states exactly in synchrony. In the transition between the two states shown above, all three switches change state. In the brief period while all are changing, the switches will read some spurious position. Even without key bounce, the transition might look like 011 — 001 — 101 — 100. When the switches appear to be in position 001, the observer cannot tell if that is the "real" position 001, or a transitional state between two other positions. 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 this problem by changing only one switch at a time, so there is never any ambiguity of position.

Binary Code	b[3:0]	g[3:0]	Gray Code
	0 0 0 0	0 0 0 0	
	0 0 0 1	0 0 0 1	
	0 0 1 0	0 0 1 1	
	0 0 1 1	0 0 1 0	
	0 1 0 0	0 1 1 0	
	0 1 0 1	0 1 1 1	
	0 1 1 0	0 1 0 1	
→	0 1 1 1	0 1 0 0	←
	1 0 0 0	1 1 0 0	
	1 0 0 1	1 1 0 1	
	1 0 1 0	1 1 1 1	
	1 0 1 1	1 1 1 0	
	1 1 0 0	1 0 1 0	
	1 1 0 1	1 0 1 1	
	1 1 1 0	1 0 0 1	
	1 1 1 1	1 0 0 0	

From the above data it can be observed that in gray code for change in output there is only one bit variation.

The output from the encoder block is send to binary to gray converter and the flits are converted to gray code and transmitted. A gray to binary converter is implemented before the flits are passed to the decoder block and the original output is retrieved back after decoding. The output of the decoder is injected into the receiver.

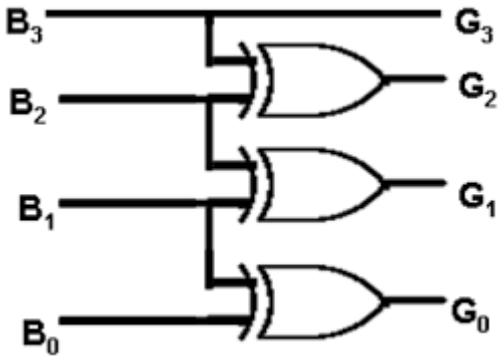
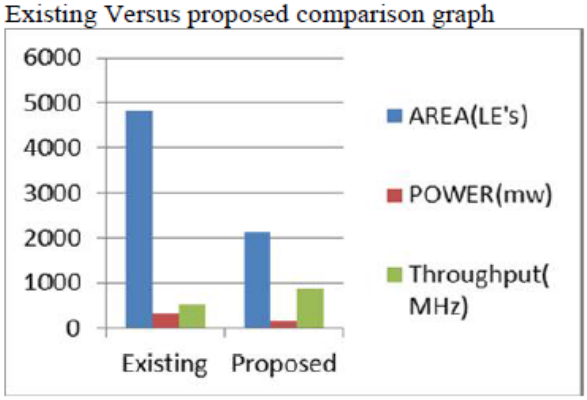


Fig. 4. A parallel binary to gray converter

IV. RESULTS

The encoder and the decoder have been designed in VHDL described at the RTL level, synthesized with Xilinx ISE Design Compiler. Here we compare the area, power and timing figures of the proposed encoding technique (SCS) against the bus-invert (BI) coding and the coupling driven bus invert (CDBI) coding as they have the highest potential for power saving while still represent a feasible implementation for on-chip communication.

Existing Versus proposed comparison graph



V. CONCLUSION

In this paper, we have presented a new data encoding scheme aimed at reducing power dissipated by the links of a NoC. Maximum power dissipation is happening in the links of the communication system. In addition, their contribution is expected to increase in future technology nodes. When compared with the existing encoding schemes, the proposed scheme will minimize not only the self switching activity, but also the coupling switching activity which is mainly responsible for link power dissipation. The proposed encoding scheme does not require any modification in routers and link architecture. Usage of gray codes for transmission of data bits leads to a high amount of reduction in power dissipation compared to existing encoding schemes. An extensive evaluation has been carried out to assess the impact of the encoder and decoder logic in the NI. 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.

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 poweraware network-on-chip (NoC)," *Integr. VLSI J.*, vol. 42, no. 4, pp. 479–485, Sep. 2009.
[6] D. Yeh, L. S. Peh, S. Borkar, J. Darringer, A. Agarwal, andW. M. Hwu, "Thousand-core chips roundtable," *IEEE Design Test Comput.*, vol. 25, no. 3, pp. 272–278, May–Jun. 2008.
[7] A. 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.
[8] 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.
[9] 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.
[10] 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.

[11] 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.
[12] 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.
[13] 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.
[14] C. 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.
[15] 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.
[16] E. 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.
[17] 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.
[18] 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.
[19] 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.
[20] G. Ascia, V. Catania, M. Palesi, and A. Parlato, "Switching activity reduction in embedded systems: A genetic bus encoding approach," *IEE Proc. Comput. Digit. Tech.*, vol. 152, no. 6, pp. 756–764, Nov. 2005.
[21] 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.
[22] 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.
[23] 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.
[24] C. 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.
[25] 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.
[26] 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.
[27] 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.
[28] 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.
[29] 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.
[30] C. 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.

