Implementation of Data Encoding Techniques for Reducing Area, Power Consumption in Network-on-Chip for LDPC Applications

Vijaykumar Jadhav¹, K. Sujata²

¹ PG Student, Department of ECE, Shree Ramchandra College of Engineering, Lonikand, Pune, India
² Professor, Department of ECE, Shree Ramchandra College of Engineering, Lonikand, Pune, India

Abstract: As technology improves, the power dissipated by the links of a network-on-chip (NoC) starts to compete with the power dissipated by the other elements of the communication subsystem, namely, the routers and the network interfaces (NIs). Here, we present a set of data encoding schemes to reduce the power dissipated by the links of a NoC. In this paper, the encoder in LDPC is replaced with our data encoding schemes in order to reduce the power consumption in Low Density Parity Check Techniques. Experiments carried out on both synthetic and real traffic scenarios show the effectiveness of the proposed schemes.

Keywords: Data Encoding, Interconnection On Chip, Low Density Parity Check, Majority Logic Decoding, Power Analysis

1. Introduction

The data encoding techniques are developed to reduce the power consumption caused by the transitions in the interconnect on the chip. The data encoding techniques are based on reducing the number of transitions by considering the types of transitions in the interconnects and by considering them as discussed in the table below and also consider the transitions as different types of inversions available for us. The different types of inversions available for us are odd inversion, full inversion and even inversions. By reducing these inversions we can control the number of transitions in the interconnects which reduces the power consumption caused by these transitions in the links.

Table I: Different Types of Inversions

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

By assuming these we have designed three schemes which are as follows:

A. Scheme I

In scheme I, we focus on reducing the numbers of Type I transitions (by converting them to Types III and IV transitions) and Type II transitions (by converting them to Type I transition). The scheme compares the current data with the previous one to decide whether odd inversion or no inversion of the current data can lead to the link power reduction.

Figure 1: Encoder architecture scheme I. (a) Circuit diagram [27] and Internal view of the encoder block.

B. Scheme II

In the proposed encoding scheme II, we make use of both odd (as discussed previously) and full inversion. The full inversion operation converts Type II transitions to Type IV transitions. The scheme compares the current data with the previous one to decide whether the odd, full, or no inversion of the current data can give rise to the link power reduction.

By assuming these we have designed three schemes which are as follows:

Volume 5 Issue 3, March 2016

www.ijsr.net

Licensed Under Creative Commons Attribution CC BY

Paper ID: NOV162344

2240
C. Scheme III

In the proposed encoding Scheme III, we add even inversion to Scheme II. The reason is that odd inversion converts some of Type I (T1***) transitions to Type II transitions. As can be observed from Table II, if the flit is even inverted, the transitions indicated as T** 1 / T1*** in the table are converted to Type IV/Type III transitions. Therefore, the even inversion may reduce the link power dissipation as well. The scheme compares the current data with the previous one to decide whether odd, even, full, or no inversion of the current data can give rise to the link power reduction.

2. Low Density Parity Check Code

The LDPC code is based on a set of one or more fundamental LDPC codes. Each of the fundamental codes is a systematic linear block code. The fundamental codes can accommodate various code rates and packet sizes. Each LDPC code in the set of LDPC codes is defined by a matrix $H$ of size $m$-by-$n$, where $n$ is the length of the code and $m$ is the number of parity check bits in the code. The number of systematic bits is $k=n-m$. The matrix $H$ is defined as

$$
H = \begin{bmatrix}
P_{1,1} & P_{1,2} & \cdots & P_{1,k-1} \\
\vdots & \vdots & \ddots & \vdots \\
P_{m,1} & P_{m,2} & \cdots & P_{m,k-1}
\end{bmatrix}
$$

Where, $P_{ij}$ is one of a set of $z$-by-$z$ permutation matrices or a $z$-by-$z$ zero matrix. The encoding of a packet at the transmitter generates parity-check bits $p=(p_0, \ldots, p_{m-1})$ based on an information block $s=(s_0, \ldots, s_{k-1})$, and transmits the parity-check bits along with the information block. Because the current symbol set to be encoded and transmitted is contained in the transmitted codeword, the information block is also known as systematic bits. The encoder receives the information block $s=(s_0, \ldots, s_{k-1})$ and uses the matrix $H_{mn}$ to determine the parity-check bits. The expanded matrix $H$ is determined from the model matrix $H_{mn}$. Since the expanded matrix $H$ is a binary matrix, encoding of a packet can be performed with vector or matrix operations conducted over GF.

3. Data Encoding Techniques Using LDPC

These data encoding schemes which are discussed in the above sections are replaced by these encoding schemes instead of the encoder in the LDPC block: A. Scheme I...
encoding technique in LDPC The proposed encoding architecture, which is based on the odd invert condition defined is shown in Fig. 1. We consider a link width of 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. This encoding technique which is scheme 1 of our encoding techniques in used in low density parity check of majority logic decoding are replaced with our encoding scheme1 technique. B. Scheme2 encoding technique in LDPC The principles of this encoder are similar to those of the encoder implementing Scheme I. The proposed encoding architecture, which is based on the odd invert condition and the full invert condition, is shown in Fig. 2. Here again, the wth bit of the previously encoded body flit is indicated with inv which defines if it was odd or full inverted (inv = 1) or left as it was (inv = 0). In this encoder, in addition to the Ty block in the Scheme I encoder, we have the T2 and T4** blocks which determine if the inversion based on the transition types T2 and T4** should be taken place for the link power reduction. The second stage is formed by a set of 1s blocks which count the number of transitions that odd inverting of pair bits leads to the link power reduction. The middle 1s block identifies the number of transitions whose full inverting of pair bits leads to the link power reduction. Finally, the bottom 1s block specifies the number of transitions whose full inverting of pair bits leads to the increased link power. The encoding technique discussed above which is scheme2 which is based on the logic of full inversion which is the proposed method of the odd inversion which is scheme1. The technique which is mentioned above is ldpc using scheme1 now is replaced with scheme2 in the ldpc and thereby reducing some amount of power compared from scheme2. Hence, we analyzed that scheme3 is the one which is used in LDPC technique.

4. Results

![Figure 5: Power report for scheme1](image1)

![Figure 6: Area report for scheme1](image2)

![Figure 7: Power report for scheme2](image3)

![Figure 8: Area report for scheme2](image4)
5. Conclusion

In this paper, we have presented data encoding techniques which are used in the place of encoders in LDPC which reduces the power consumption by eliminating the transitions as discussed before. Here, we analyzed the power consumption for these three schemes and compared their power and area performances.

References

[1] International Technology Roadmap for Semiconductors. (2011).

[2] M. S. Rahaman and M. H. Chowdhury, “Crosstalk avoidance and error correction 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] Pedro Reviriego, Juan A. Maestro, and Mark F. Flanagan, “Error Detection in Majority Logic Decoding of Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 1, pp.156-159, Jan. 2013.

[5] Youn Sung Park, Yaoyu Tao, Zhengya Zhang, “A 1.15Gb/s Fully Parallel Nonbinary LDPC Decoder with Fine-Grained Dynamic Clock Gating,” 2013 IEEE International Solid-State Circuits Conference.

[6] L. Macchiarulo, E. Macii, and M. Poncino, —Wire placement for crosstalk energy minimization in address buses,l in Proc. Design Autom.Test Eur. Conf. Exhibit., Mar. 2002, pp. 158–162.

[7] R. Ayoub and A. Orailoglu, —A unified transformational approach for reductions in fault vulnerability, power, and crosstalk noise and delay on processor buses,l in Proc. Design Autom. Conf. Asia South Pacific, vol. 2. Jan. 2005, pp. 729–734.

[8] K. Banerjee and A. Mehrotra, —A power-optimal repeater insertion methodology for global interconnects in nanometer designs,l IEEE Trans. Electron Devices, vol. 49, no. 11, pp. 2001–2007, Nov. 2002.

[9] M. R. Stan and W. P. Burleson, —Bus-invert coding for low-power I/O,l IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 3, no. 1, pp. 49–58, Mar. 1995.

[10] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, —A coding framework for low-power address and data busses,l IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 7, no. 2, pp. 212–221, Jun. 1999.