Design and Analysis of Effective Data Encoding Techniques for Parallel Links in NOC

S.Sree Divya Teja
Tejasomesula@Gmail.Com
Department Of Ece Malineni Lakshmaiah Engineering College
G.Suresh
Associate Professor Department Of Ece Malineni Lakshmaiah Engineering College
Gsuresh47@Gmail.Com

ABSTRACT:

A system-on-chip (SoC) combines the required electronic circuits of various computer components onto a single, integrated chip (IC). SoC is a complete electronic substrate system that may contain analog, digital, mixed-signal or radio frequency functions. As the density of VLSI design increases, the complexity of each component in a system raises rapidly. Today's SoC designers face a new challenge in the design of the on-chip interconnects beyond the evolution of an increasing number of processing elements. The main problems in SoC are wire delays, synchronization, uncertainity, power etc. hence to overcome these problems we go for Network on Chip (NoC). In the case of large-scale designs, network on chip is preferred as it reduces the complexity involved in designing the wires and also provides a well-controlled structure capable of better power, speed and reliability. For high-end system-on-chip designs, network-on-chip (NoC) is considered the best integrated solution. As the technology shrinks, the power dissipated by the links of a NoC starts to compete with the power dissipated by the other elements of the communication subsystem, namely, the routers and the network interfaces (NI). In this project, I analyzed a set of data encoding schemes aimed at reducing the power dissipated by the links of a NoC with small area overhead. The proposed schemes are general and transparent with respect to the underlying NOC fabric.

I. INTRODUCTION

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 project, 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 [9]. 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.

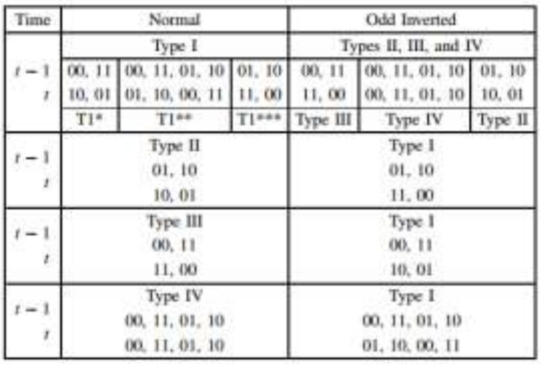
The rest of this paper is organized as follows. We briefly discuss related works in Section II, while Section III presents an overview of the proposed data encoding schemes. The proposed data encoding schemes along with possible hardware implementations and their analysis are described in Section IV. 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.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 18
December2016

II. 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 aforementioned 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]–[30]. 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 [29] and [30] 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 [30], the coupling switching activity is reduced up to 39%. In this paper, compared to [30], we use a simpler decoder while achieving a higher activity reduction. 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.

TABLE I
Effect of Odd Inversion on Change of Transition Types



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		

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 three encoding schemes. In Scheme I, we focus on reducing Type I transitions while in Scheme II, both Types I and II transitions are taken into account for deciding between half and full invert, depending the amount of switching reduction. Finally, in Scheme III, we consider the fact that Type I transitions show different behaviours in the case of odd and even inverts and make the inversion which leads to the higher power saving.

III. 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].

IV. 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. Let us first describe the power model that contains different components of power dissipation of a link. The dynamic power dissipated by the interconnects and drivers is

$P = [T_{0\to1} (C_s + C_l) + T_cC_c]\ V_{dd}^{2}\ F_{ck}$ (1)

where $T_{0\to1}$ is the number of 0 → 1 transitions in the bus in two consecutive transmissions, T_c is the number of correlated switching between physically adjacent lines, C_s is the line to substrate capacitance, C_l is the load capacitance, C_c is the coupling capacitance, V_{dd} is the supply voltage, and F_{ck} is the clock frequency. One can classify four types of coupling transitions as described in [26]. A Type I transition occurs when one of the lines switches when the other remains unchanged. In a Type II transition, one line switches from low to high while the other makes transition from high to low. A Type III transition corresponds to the case where both lines switch simultaneously. Finally, in a Type IV transition both lines do not change. The effective switched capacitance varies from type to type, and hence, the coupling transition activity, T_c, is a weighted sum of different types of coupling transition contributions [26]. Therefore

$T_c = K_1T_1 + K_2T_2 + K_3T_3 + K_4T_4$ (2)

where T_i is the average number of Type i transition and K_i is its corresponding weight. According to [26], we use $K_1 = 1$, $K_2 = 2$, and $K_3 = K_4 = 0$. The occurrence probability of Types I and II for a random set of data is 1/2 and 1/8, respectively. This leads to a higher value for K_1T_1 compared with K_2T_2 suggesting that minimizing the number of Type I transition may lead to a considerable power reduction. Using (2), one may express (1) as

$P = [T_{0\to1} (C_s + C_l) + (T_1 + 2T_2)\ C_c]\ V_{dd}^{2}\ F_{ck}$. (3)

According to [3], C_l can be neglected

$P \propto T_{0\to1}C_s + (T_1 + 2T_2)\ C_c$. (4)

Here, we calculate the occurrence probability for different types of transitions. Consider that flit (t - 1) and flit (t) refer to the previous flit which was transferred via the link and the flit which is about to pass through the link, respectively. We consider only two adjacent bits of the physical channel. Sixteen different combinations of these four bits could occur (Table I). Note that the first bit is the value of the generic ith line of the link, whereas the second bit represents the value of its (i + 1)th line. The number of transitions for Types I, II, III, and IV are 8, 2, 2, and 4, respectively. For a random set of data, each of these sixteen transitions has the same probability. Therefore, the occurrence probability for Types I, II, III, and IV are 1/2, 1/8, 1/8, and 1/4, respectively. In the rest of this section, we present three data encoding schemes designed for reducing the dynamic power dissipation of the network links along with a possible hardware implementation of the decoder.

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.

1) Power Model: If the flit is odd inverted before being transmitted, the dynamic power on the link is $P \propto T_{0\to1} + K_1T_1 + K_2T_2 + K_3T_3 + K_4T_4\ C_c$ (5) where $T_{0\to1}$, T_1, T_2, T_3, and T_4, are the self-transition activity, and the coupling transition activity of Types I, II, III, and IV, respectively. Table I reports, for each transition, the relationship between the coupling transition activities of the flit when transmitted as is and when its bits are odd inverted. Data are organized as follows. The first bit is the value of the generic *i*th line of the link, whereas the second bit represents the value of its (*i* + 1)th line. For each partition, the first (second) line represents the values at time *t* - 1 (t). As Table I shows, if the flit is odd inverted, Types II, III, and IV transitions convert to Type I transitions. In the case of Type I transitions, the inversion leads to one of Types II, III, or Type IV transitions. In particular, the transitions indicated as T_{*1}, T_{1**}, and T_{1***} in the table convert to Types II, III, and IV transitions, respectively. Also, we have $T_{0\to1} = T_{0\to0(odd)} + T_{0\to1(even)}$ where odd/even refers to odd/even lines. Therefore, (5) can be expressed as $P \propto T_{0\to0(odd)} + T_{0\to1(even)}\ C_s + [K_1(T_2+T_3+T_4) + K_2T_{1***} + K_3T_{1*} + K_4T_{1**}]C_c$. (6) Thus, if $P > P$, it is convenient to odd invert the flit before transmission to reduce the link power dissipation. Using (4) and (6) and noting that $C_c/C_s = 4$ [26].

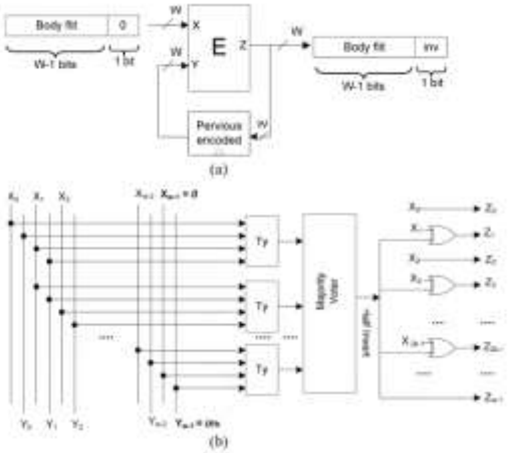


Fig. 1. Encoder architecture scheme I. (a) Circuit diagram [27]. (b) Internal view of the encoder block (E).

Also, since $T_{0\to1} = T_{0\to1(odd)} + T_{0\to1(even)}$, one may write $1\ 4\ T_{0\to1(odd)} + T_1 + 2T_2 > 1\ 4\ T_{0\to0(odd)} + T_2 + T_3 + T_4 + 2T_{1***}$ (7)

which is the exact condition to be used to decide whether the odd invert has to be performed. Since the terms $T_{0\to1(odd)}$ and $T_{0\to0(odd)}$ are weighted with a factor of 1/4, for link widths greater than 16 bits, the misprediction of the invert condition will not exceed 1.2% on average [23]. Thus, we can approximate the exact condition as $T_1 + 2T_2 > T_2 + T_3 + T_4 + 2T_{1***}$. (8) Of course, the use of the approximated odd invert condition reduces the effectiveness of the encoding scheme due to the error induced by the approximation but it simplifies the hardware implementation of encoder. Now, defining $T_x = T_3 + T_4 + T_{1***}$ and $T_y = T_2 + T_1 - T_{1***}$ (9) one can rewrite (8) as $T_y > T_x$. (10) Assuming the link width of *w* bits, the total transition between adjacent lines is *w* - 1, and hence $T_y + T_x = w - 1$. (11) Thus, we can write (10) as $T_y > (w - 1)\ 2$. (12) This presents the condition used to determine whether the odd inversion has to be performed or not.

2) Proposed Encoding Architecture:

The proposed encoding architecture, which is based on the odd invert condition defined by (12), 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. More specifically, the NI packs the body flits in *w* - 1 bits [Fig. 1(a)]. 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. The generic block diagram shown in Fig. 1(a) is the same for all three encoding schemes proposed in this paper and only the block *E* is different for the schemes. To make the decision, the previously encoded flit is compared with the current flit being transmitted. This latter, whose *w* bits are the concatenation of *w* - 1 payload bits and a “0” bit, represents the first input of the encoder, while the previous encoded flit represents the second input of the encoder [Fig. 1(b)]. The *w* - 1 bits of the incoming (previous encoded) body flit are indicated by X_i (Y_i), *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). In the encoding logic, each *Ty* block takes the two adjacent bits of the input flits (e.g., X_1 $X_2Y_1Y_2$, X_2 $X_3Y_2Y_3$, X_3 $X_4Y_3Y_4$, 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. The second stage of the encoder, which is a majority voter block, determines if the condition given in (12) is satisfied (a higher number of 1s in the input of the block compared to 0s). If this condition is satisfied, in the last stage, the inversion is performed on odd bits. The decoder circuit simply inverts the received flit when the inversion bit is high.

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.

1) Power Model: Let us indicate with *P*, *P*, and *P*

the power dissipated by the link when the flit is transmitted with no inversion, odd inversion, and full inversion, respectively. The odd inversion leads to power reduction when *P* < *P*

and *P* < *P*. The power *P*

is given by [23] *P*

$\propto T_1 + 2T_{4**}$. (13) Neglecting the self-switching activity, we obtain the condition *P* < *P*

as [see (7) and (13)] $T_2 + T_3 + T_4 + 2T_{1***} < T_1 + 2T_{4**}$. (14) Therefore, using (9) and (11), we can write $2\,T_2 - T_{4**} < 2T_y - w + 1$. (15).

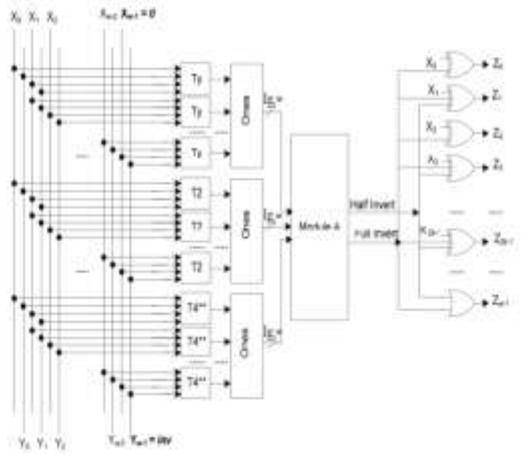


Fig. 2. Encoder architecture Scheme II.

2) Proposed Encoding Architecture: The operating 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 of (16) and the full invert condition of (18), is shown in Fig. 2. Here again, the *w*th bit of the previously and the full invert condition of (18) is shown in Fig. 2. Here again, the *w*th 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 T_2 and T_{4**} blocks which determine if the inversion based on the transition types T_2 and T_{4**} 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 1s in their inputs. The output of these blocks has the width of $\log_2 w$. The output of the top 1s block determines 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. Based on the number of 1s for each transition type, Module A decides if an odd invert or full invert action should be performed for the power reduction.

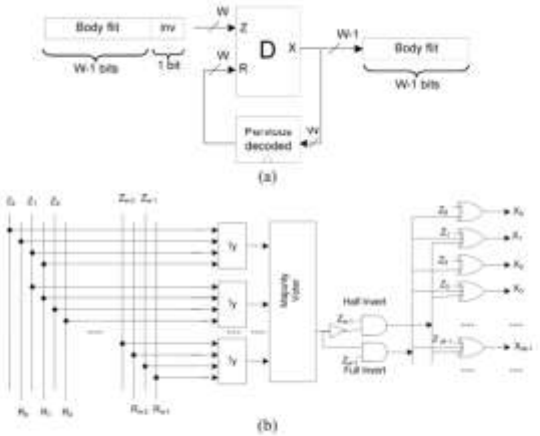


Fig. 3. Decoder architecture Scheme II. (a) Circuit diagram. (b) Internal view of the decoder block (D).

For this module, if (16) or (18) is satisfied, the corresponding output signal will become “1.” In case no invert action should be taken place, none of the output is set to “1.” Module A can be implemented using full-adder and comparator blocks. The circuit diagram of the decoder is shown in Fig. 3. The w bits of the incoming (previous) body flit are indicated by Z_i (R_i), $i = 0, 1, \ldots, w - 1$. The wth bit of the body flit is indicated by inv which shows if it was inverted (inv = 1) or left as it was (inv = 0). For the decoder, we only need to have the T_y block to determine which action has been taken place in the encoder. Based on the outputs of these blocks, the majority voter block checks the validity of the inequality given by (12). If the output is “0” (“1”) and the inv = 1, it means that half (full) inversion of the bits has been performed. Using this output and the logical gates, the inversion action is determined. If two inversion bits were used, the overhead of the decoder hardware could be substantially reduced.

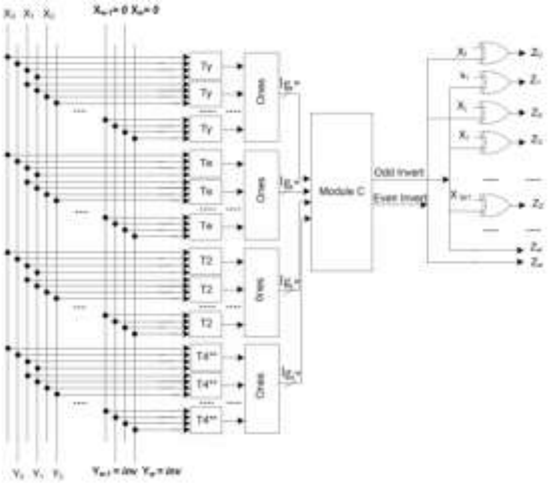
C. Scheme III

In the proposed encoding Scheme III, we add even inversion to Scheme II. The reason is that odd inversion converts some of Type I (T_{1***}) transitions to Type II transitions. As can be observed from Table II, if the flit is even inverted, the transitions indicated as T_{**1}/T_{1***} 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.

TABLE II
EFFECT OF EVEN INVERSION ON CHANGE OF TRANSITION TYPES

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

2) Proposed Encoding Architecture: The operating principles of this encoder are similar to those of the encoders implementing Schemes I and II. The proposed encoding architecture, which is based on the even invert condition of (28), the full invert condition of (29), and the odd invert condition of (30), is shown in Fig. 4. The wth bit of the previously encoded body flit is indicated by inv which shows if it was even, odd, or full 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 first stage, we have added the T_e blocks which determine if any of the transition types of T_2, 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 T_y, T_e, T_2, 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. The outputs “01,” “11,” and “10” show that whether (28), (29), and (30), respectively, are satisfied. In this paper, Module C was designed based on the conditions given in (28), (29), and (30). Similar to the procedure used to design the decoder for scheme II, the decoder for scheme III may be designed.



V. Results and Discussion

The proposed data encoding schemes have been assessed by means of a cycle-accurate NoC simulator based on Noxim [33]. The power estimation models of Noxim include NIs, routers, and links [25]. The link power dissipation was computed using (3) where the terms *T*0→1, *T*1, and *T*2 were computed based on the information obtained from the cycle accurate simulation.

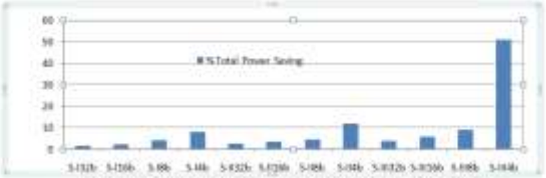


Fig.3. Total power energy saving using different data encoding schemes

The following parameters were used in the simulations. The NoC was clocked at 700 MHz while the baseline NI with minimum buffering and supporting open core protocol 2 and advanced high-performance bus protocols [34] dissipated 5.3 mW. The average power dissipated by the wormhole-based router was 5.7 mW. Based on a 65-nm UMC technology, a total capacitance of 592 fF/mm was assumed for an inter-router wire.

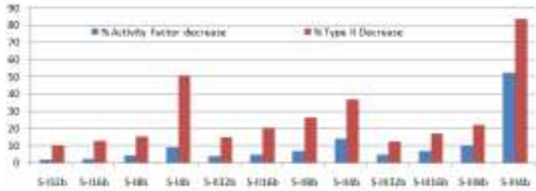
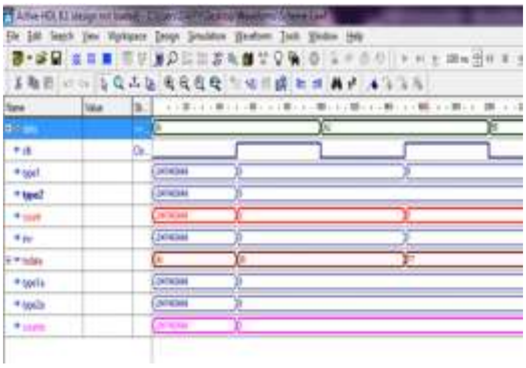


Fig.4. Percentage of decrease Types II, and coupling switching activity obtained with different data encoding

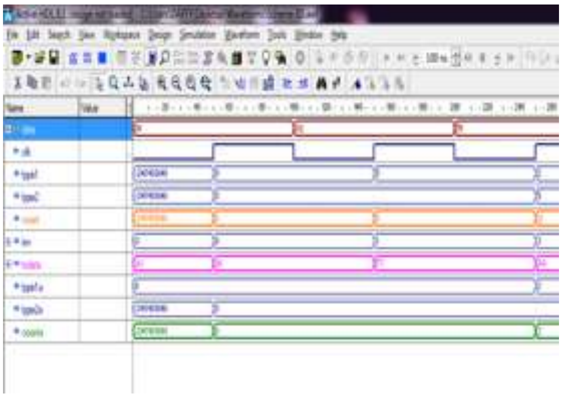
About 80% of this capacitance was due to the crosstalk. We assumed 2-mm 32-bit links and a packet size of 16 bytes (eight flits). Using the detailed simulations, when the flits traversed the NoC links, the corresponding selfand coupling switching activities were calculated and used along with the self- and coupling capacitance of 0.237 and 0.947 nf, respectively, to calculate the power (*V*dd = 0.9 V and *F*ck = 700 MHz).

No Of Bits	Power without Schemes	Power with S-I	Efficiency	Power with S-II	Efficiency	Power with S-III	Efficiency
500	4.353	4.346	0.16	4.189	3.76	4.168	4.25
1000	8.798	8.754	0.5	8.448	3.98	7.808	11.2
1500	13.12	13.03	0.6	12.66	3.5	11.87	9.52
2000	17.41	17.02	2.24	16.81	3.44	15.82	9.13
2500	21.92	21.56	1.64	21.11	3.69	19.74	9.94

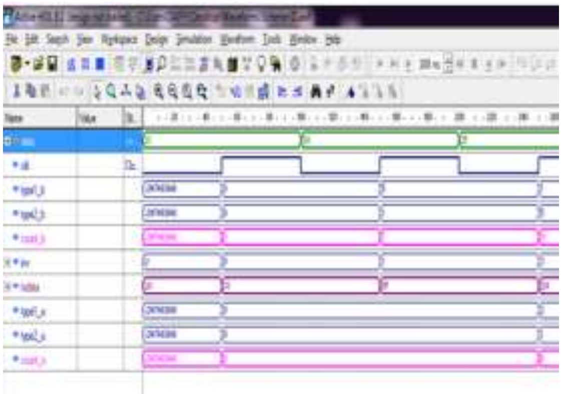
Table 6.24: Comparison of Power Results for 32-bit data


File Edit Search View Workspace Design Simulation Waveform Tools Window Help

SCHEME-I (16-BIT)


File Edit Search View Workspace Design Simulation Waveform Tools Window Help

SCHEME-II (16-BIT)


File Edit Search View Workspace Design Simulation Waveform Tools Window Help

SCHEME-III (16-BIT)

VI. CONCLUSION AND FUTURE SCOPE

A power efficient Data bus encoding method, which reduces the power dissipation with respect to both self transitions and coupling transitions, has been proposed. In the present scenario inter-connects plays a dominant role in the dynamic power consumption. These inter-connects consumes up to 50% of total dynamic power. In this method the coupling transitions have been reduced with the encoding method.

For 64-bit data of 2500 lines, we got the efficiencies as 3.85%, 8.54%, 7.69% by using Schemes I, II, III respectively and this can be increased to 47.37% and 87.12% by using shuffling codes with Schemes I and II.

Simulation results have shown that the proposed data encoding methods are suitable for the continually shrinking technology. Hence the proposed coding schemes are suitable for energy VLSI applications.

As the technology enters into nanometre technology, the effect of the crosstalk plays an important role on the reliability of transferring the data on data bus. We can achieve better performance by combining this proposed codec with Error Correcting Codes (ECC). The encoder and decoder adds to extra hardware overhead. This codec power consumption schemes should be considered to evaluate all power savings. To increase the efficiency further, this codec can be extended by using hamming code.

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 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 MPSoCtechnology,” *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. 199.