

Journal homepage: http://www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH

RESEARCH ARTICLE

Realization of Data Encoding and Decoding Scheme for NOC Application

Shaik. Mahaboob Basha, D. Sri Hari, N. Pavitra

Manuscript Info	Abstract
Manuscript History:	Network-on-chip (NoC) topology is composed by an arbitrary number of
Received: 15 March 2015 Final Accepted: 28 April 2015 Published Online: May 2015	and Switches. In NoC the major source for power dissipation is the NoC links. The self and coupling switching activities are responsible for link
Key words:	decoding schemes operating at flit level and on an end-to-end basis, which allows us to minimize both self and coupling switching activities. The self-
Data encoding, NoC, Links, Low power, Coupling switching activity, End-to-End technique.	switching is reduce by checking the switching transition and then the coupling technique is incorporated with the wormhole routed network, that is flits are appended by the network interface before they are injected in the
*Corresponding Author	network and are decoded by the destination NI (network interface). Especially the decoding schemes are focused on reducing hardware. This
Shaik. Mahaboob Basha	paper analyzes the power and delay reduction of both encoding and decoding schemes and also analyzes the 3-bit odd and even inversion transition types in Xilinx Virtex-5 family (XC5VLX50).
	Copy Right, IJAR, 2015,. All rights reserved

INTRODUCTION

NoC architectures popularity comes directly from the growing interest around System-on-Chip (SoC) and Multi-Processor-System-on-Chip (MPSoC) technologies. NoC technology is a "Front end solution to a back end problem". NoC is a technology of integrating a complex network system into a single chip. Company like 'Arteries' provides NoC interconnect semiconductor intellectual property to SOC which can easily reduce cycle time, increase margin and can improve functionality on SoC smoothly. As by increasing the market demand the researchers have added more number of traditional hierarchal buses or crossbar to SOC which becomes very complex for which the architecture and integration is affected. Due to large number of wires interconnection heat and power consumption is increased. These problems have one solution which is NoC. Network-on-chip helps to reduce SOC manufacturing cost, increase performance of SOC, reduce SOC time consumption and also reduce the SOC design risk [1].

Network-on-chip topology is composed by an arbitrary number of instances of three basic kind of functional blocks Network Interfaces, Switches and Links. Network Interfaces connect all the IP cores to the network. Switches carry out the task of dispatching packets inside the network, depending on the particular routing scheme and links connect switches with Nis. The Network on chip architecture is shown in the Fig.1.



Fig.1 Network on chip architecture

BUS	NoC				
250MHz	>750MHz				
9GB/s	100GB/s				
(more if wider bus)					
5GB/s	100GB/s				
(more if wider bus)					
400K	210K				
Smaller for NoC					
Smaller for NoC					
(proportional to gate count)					
	BUS 250MHz 9GB/s (more if wider bus) 5GB/s (more if wider bus) 400K Smaller for Smaller for (proportional to				

The comparison between traditional bus and NoC is shown in the below Table.1.

Table.1 comparison between traditional bus and NoC

The Table.1 shows that for designs of the complexity level that we used for the comparison, the NoC approach has a clear advantage over traditional busses for nearly all criteria, most notably system throughput.

In this paper we have focused on power dissipated by the NoC links. In fact, the power dissipated by the network links is as relevant as that dissipated by routers and network interfaces (NIs) and their contribution is expected to increase as technology scales [2]. The self-switching and coupling switching activities are responsible for link power dissipation. So we present a set of Data encoding and Data decoding schemes operating at flit level and on an end-to-end basis, which allows us to minimize both the self and coupling switching activities on links of the routing paths traversed by the packets. The proposed data encoding and decoding schemes, which are transparent with respect to the router implementation, are presented and discussed at both the algorithmic level and the architectural level. The analysis takes into account several aspects and metrics of the design, including silicon area, delay, power dissipation and energy consumption.

The rest of this paper is organized as follows. We briefly discuss related works in Section II. The proposed data encoding and decoding schemes along with possible hardware implementations and their analysis are described in Section III. In Section IV proposed data encoding and decoding schemes are simulated and verified using verilog HDL in Xilinx ISE 10.1i for the target device xc3s500e-5fg320. Finally, this paper is concluded in Section V.

II.RELATED WORKS AND CONTRIBUTIONS

The basic idea is to encode the data before traversing through links to reduce the switching activity of the links. In this proposal, the data is encoded before transmission and is decoded at the destination. The NI is augmented with an encoder (E) and a decoder (D) block. The encoder encodes the outgoing flits of the packet in such a way as to minimize the power dissipated by the inter router point-to-point links which form the routing path of the current packet. Since the routers are not equipped with any encoding/decoding logic, the header flit is not encoded as it contains control information (packet size, destination address, etc.) which have to be processed by the routers through the routing path. The decoder decodes the incoming flits in the NI. The General Scheme for Proposed approach is shown in the below Fig.2.



Fig.2 General Scheme for Proposed approach

This paper concentrates on reducing the power dissipated by the links. The data encoding scheme is one of the major method that was employed to reduce the link power dissipation. The data encoding techniques may be

classified into two categories. In the first category Bus invert (BI) and INC-XOR have been proposed for the case that random data patterns are transmitted via these lines. In this category, Bus invert (BI) encoding method is widely used technique to reduce dynamic switching power. The basic idea behind Bus invert encoding method originated by noting that a lot of power is wasted during data transmission in off-chip bus lines. This is due to the switching of the high capacitance lines, therefore power could be saved by minimizing the number of transitions occurring on these bus lines. This method includes two values data value and bus value. The data value is denoted as a piece of information that has to be transmitted over the bus in a given time-slot and the bus value will denote the coded value. Typically a code needs extra control bits, so BI method uses one extra control bit called "invert". The invert = 0 the bus value will equal the data value. When invert = 1 the bus value will be the inverted data value [3] is shown in the below Fig.3.

Current Simulation Time: 1000 ns		900 n I	IS I	1	910 I	ns I	I		92 	20 ns		93 	0 ns 1	1	94 	0 ns	1	95 	0 ns	1
🎝 invert	1																			
🖬 🚮 data_out[8:0]	9																9	b110	0101	101
🖬 🚮 count(3:0)	4'h6																	4	'h6	
🗖 😽 data(7:0)	8																8	'b10	0101	01

Fig.3 Bus-invert Encoding method

So in the Bus-invert method peak power dissipation is reduced by half and in order to decrease the number of transitions.

On the other hand gray code, working-zone encoding and T0-XOR were suggested for the case of correlated data patterns. In this second category, gray code encoding method is widely used technique to reduce power dissipation by minimizing the coupling transition activities on the links of the inter connection network and to reduce errors using gray code. The basic idea behind gray code encoding method of this approach is encoding the flits before they are injected into the network with the goal of minimizing the self switching and coupling switching activities in the links traversed by the flits. It consists of a binary to gray converter, encoder, decoder and gray to binary converter. 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] is shown in the below Figures.

Current Simulation Time: 1000 ns		900 ns 1	3	910 ns			920 ns		930 ns		940 ns		950 ns
🖬 🚮 B[3:0]	4												4'b1000
🖬 🚮 G[3:0]	4												4101100
		Fig	g.4	bina	ry t	0	gray	co	nversi	on			
Current Simulation Time: 1000 ns		900 ns	;	910 ns			920 ns		930 ns		940 ns		950 ns
🖬 😽 G[3:0]	4												411100
a 🔊 B(3:01	1												464000

Fig.5 gray to binary conversion

The self switching activity and coupling switching activity are responsible for link power dissipation [5]. So we refer to the end-to-end scheme and this end-to-end encoding scheme takes advantage of the pipeline nature of the wormhole switching technique [6].

In this paper, we present three proposed encoding schemes. In scheme I, we make use half inversion and focus on reducing Type I transitions. 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 behaviors in the case of odd and even inverts [7] and make the inversion which leads to the higher power saving.

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

TABLE I EFFECT OF ODD INVERSION ON CHANGE OF TRANSITION TYPES

III.PROPOSED ENCODING AND DECODING SCHEMES

In this section, we present the proposed encoding scheme and decoding 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

$$P = [T_{0\to 1} (C_{\rm s} + C_l) + T_{\rm c} C_{\rm c}] V_{\rm dd}^2 F_{\rm ck}$$
(1)

Where $T_{0\to 1}$ is the number of $0\to 1$ 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, $C_{\rm c}$ is the coupling capacitance, $V_{\rm dd}$ is the supply voltage, and $F_{\rm ck}$ is the clock frequency. One can classify four types of coupling transitions as described in [8]. 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 [8]. Therefore

$$T_{\rm c} = K_1 T_1 + K_2 T_2 + K_3 T_3 + K_4 T_4 \tag{2}$$

Where T_i is the average number of Type *i* transition and K_i is its corresponding weight. According to [8], we use $K_1=1$, $K_2=2$, and $K_3=K_4=0$. The occurrence 1/2 and 1/8, respectively. This leads to a higher value for K_1T compared with K_2T suggesting that minimizing the number of TypeI transition may lead to a considerable power reduction. Using (2), one may express (1) as

$$P = [T_{0 \to 1} (C_{\rm s} + C_l) + (T_1 + 2T_2) C_{\rm c}] V_{\rm dd}^2 F_{\rm ck}$$
(3)

According to [3], C_l can be neglected

$$P \alpha T_{0 \to 1} C_{\rm s} + (T_1 + 2T_2) C_{\rm c} \tag{4}$$

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 *i*th 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

$$P \propto T'_{0 \to 1} + (K_1 T'_1 + K_2 T'_2 + K_3 T'_3 + K_4 T'_4)C_c$$
(5)

Where $T'_{0\to 1}$, 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 relation-ship 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 $T'_{0\to 1} = T_{0\to 0(\text{odd})} + T_{0\to 1(\text{even})}$ where odd/even refers to odd/even lines. Therefore, (5) can be expressed as

$$P \alpha (T_{0 \to 0(\text{odd})} + T_{0 \to 1(\text{even})})C_{\text{s}} + [K_1 (T_2 + T_3 + T_4) + K_2 T_1^{***} + K_3 T_1^* + K_4 T_1^{**}]C_{\text{c}}$$
(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 $C_c/C_s = 4$ [7], we obtain the following odd invert condition

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

$$\frac{1}{4}T_{0\to 1(\text{odd})} + T_1 + 2T_2 > \frac{1}{4}T_0 \rightarrow 0(\text{odd}) + T_2 + T_3 + T_4 + 2T_1^{***}$$
(7)

Which is the exact condition to be used to decide whether the odd invert has to be performed. Since the terms $T_{0\to 1(\text{odd})}$ and $T_{0\to 0(\text{odd})}$ 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. 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

 $T_{\rm x} = T_3 + T_4 + T_1^{***}$

and

 $T_{\rm y} = T_2 + T_1 - T_1^{***} \tag{9}$

One can rewrite (8) as

$$T_{\rm v} > T_{\rm x} \tag{10}$$

Assuming the link width of w bits, the total transition between adjacent lines is w - 1, and hence $T_v + T_x = w - 1$ (11)

Thus, we can write (10) as

$$T_v > (w - 1)/2$$
 (12)

This presents the condition used to determine whether the odd inversion has to be performed or not.



Fig.6 Encoder architecture scheme I (a) Circuit diagram (b) Internal view of the encoder block (E)

2) Proposed Encoding Architecture: The proposed encoding architecture, which is based on the oddinvert condition defined by (12), is shown in Fig.6. 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. 6(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. 6(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. 6(b)]. The w-1 bits of the incoming (previous encoded) body flit are indicated by Xi (Yi), $i = 0, 1, \ldots, w$ - 2. The wth 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., $X_1X_2Y_1Y_2$, $X_2X_3Y_2Y_3$, $X_3X_4Y_3Y_4$, 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

$$P'' \propto T_1 + 2T_4^{**} \tag{13}$$

Neglecting the self-switching activity, we obtain the condition P' < P'' as

 $T_2 + T_3 + T_4 + 2T_1^{***} < T_1 + 2T_4^{**}$ (14)

Therefore, using (9) and (11), we can write

$$2(T_2 - T_4^{**}) < 2T_y - w + 1 \tag{15}$$

Based on (12) and (15), the odd inversion condition is Obtained as

$$2(T_2 - T_4^{**}) < 2T_y - w + 1 \quad T_y > (w - 1)/2 \tag{16}$$

Similarly, the condition for the full inversion is obtained from P'' < P and P'' < P'. The inequality P'' < P is satisfied when

$$T_2 > T_4^{**}$$
 (17)

Therefore, using (15) and (17), the full inversion condition is obtained as $2(T_2 - T_4^{**}) > 2T_y - w + 1$ $T_2 > T_4^{**}$ (18)

When none of (16) or (18) is satisfied, no inversion will be performed.



Fig.7 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. 7. Here again, the *w*th bit of the previously and the full invert condition of (18) is shown in Fig. 7. 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 log2 *w*. The output of the top 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 link power reduction. Finally, the bottom 1s block specifies 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.

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. 8. The w bits of the incoming (previous) body flit are indicated by Zi (Ri), i = 0, 1, ..., 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 Ty 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.



Fig.8 Decoder architecture Scheme II (a) Circuit diagram (b) Internal view of the decoder block (D)

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.

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, full inversion, and even inversion, respectively. Similar to the analysis given for Scheme I, we can approximate the condition P'' < P as

$$T_1 + 2T_2 > T_2 + T_3 + T_4 + 2T_1^*$$
TABLE II
(19)

EFFECT OF EVEN INVERSION ON CHANGE OF TRANSITION TYPES

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

Defining

$$T_{\rm e} = T_2 + T_1 - T_1^{*} \tag{20}$$

We obtain the condition $P^{''} < P$ as

 $T_{\rm e} > (w - 1)/2$ (21) Similar to the analysis given for scheme II, we can approximate the condition $P^{,,,} < P^{,}$ as

$$T_2 + T_3 + T_4 + 2T_1^* < T_2 + T_3 + T_4 + 2T_1^{***}$$
(22)

Using (9) and (20), we can rewrite (22) as

$$T_{\rm e} > T_{\rm y} \tag{23}$$

Also, we obtain the condition $P^{"} < P"$ as [see (13) and (19)]

$$T_2 + T_3 + T_4 + 2T_1 < T_1 + 2T_4^{**}$$
(24)

Now, define

$$T_{\rm r} = T_3 + T_4 + T_1^*$$

and

$$T_{\rm e} = T_2 + T_1 - T_1^{*}$$
(25)

Assuming the link width of w bits, the total transition between adjacent lines is w - 1, and hence

$$T_{\rm e} + T_{\rm r} = w - I \tag{26}$$

Using (26), we can rewrite (24) as

$$2(T_2 - T_4^{**}) < 2T_e - w + 1$$
(27)

The even inversion leads to power reduction when P' < P, P' < P', and P' < P'''. Based on (21), (23), and (27),

We obtain

$$T_{\rm e} > (w-1)/2, T_{\rm e} > T_{\rm v}, 2(T_2 - T_4^{**}) < 2T_{\rm e} - w + 1$$
 (28)

The full inversion leads to power reduction when P'' < P, P'' < P', and P'' < P'''. Therefore, using (18) and (27), the full inversion condition is obtained as

2($T_2 - T_4^*$) $2T_y - w + 1, (T_2 > T_4^{**})$

$$2(T_2 - T_4^*) > 2T_e - w + 1 \tag{29}$$

Similarly, the condition for the odd inversion is obtained from P < P, P < P, and P < P. Based on (16) and (23), the odd inversion condition is satisfied when

$$2(T_2 - T_4^{**}) < 2T_y - w + 1, T_y > (w - 1)/2$$

$$T_e < T_y$$
(30)

When none of (28), (29), or (30) is satisfied, no inversion will be performed.

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.9. 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 transitions types while the second stage is formed by set of 1s blocks which counts 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 which determine if any of the 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). Similar to the procedure used to design the decoder for scheme III designed shown in the below Fig.10.



Fig.10 Decoder internal view of scheme III

IV. RESULTS AND DISCUSSION

The proposed Encoding and Decoding schemes include four types of coupling transitions. 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.

2-variable coupling transitions:

	Гур	e I		
00	01	10	11	
01	11	00	01	
10	00	11	10	
]	Гуре	e II		
)1	10		
1	0	01		
Т	`ype	eIII		
)0	11		
]	1	00		
Т	'ype	e IV		
00	01	10	11	
00 ()1	10	11	

Type I:

Current Simulation Time: 1000 ns		0 ns 11	0 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns
g Z	1									
= 🕅 x(1:0]	2'n0									
<mark>ð,[</mark> x[1]	0									
<mark>ð [</mark> x[0]	0									
■ <mark>61</mark> y(1:0]	2'h1	2110					2ħ1			
<mark>ð,[</mark> y[1]	0									
<mark>ð,1</mark> y(0)	1									

Type II:

Current Simulation Time: 1000 ns		0 ns 10	0 ns 200 ns 300 ns 400 ns 500 ns 800 ns 700 ns 800 n N N N N N N N N N N N N N N N N N N N	ns I
ğ, Z	1			
■ <mark>Ş1</mark> x(1:0]	21h2	2h0	2h2	
<mark>6,</mark> [1(1]	1			
<mark>6,1</mark> ×[0]	0			
■ <mark>ŞN</mark> y(1:0]	2111	2h0	2h1	
<mark>6,1</mark> v(1)	0			
<mark>6,1</mark> y[0]	1			

Type III:

							15.9 ns				
Current Simulation Time: 1000 ns		0 ns 10	Ins IIIII	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns	900 ns 1000 n
à Z	1										
■ <mark>81</mark> x(1:0]	2113	2h0					25				
<mark>3</mark> []#[1]	1										
<mark>)</mark> (I)	1										
■ <mark>Şi</mark> y(10]	210										
<mark>3.</mark> [y[1]	0										
<mark>ð í</mark> yll	0										

								662	2 กร			
Current Simulation Time: 1000 ns		0 ns 10 	10 ns	200 ns	300 ns	400 ns	500 ms	600 ns	700 ns	800 ns	900 	ns 1000
2 Z	1											
■ <mark>Ş\[</mark> x[1:0]	2h1	2h0	χ									
<mark>3</mark> [x[1]	0											
<mark>)</mark> [x[0]	1											
■ <mark>\$1</mark> y[1:0]	2h1	2110	X									
<mark>31</mark> y(1)	0											
<mark>8</mark> 40	1											

Type IV:

3-variable coupling transitions:

			Ту	vpe I			
000	001	010	011	100	101	110	111
001	101	110	111	110	111	100	011
010	011	000	001	101	001	111	101
100	000	011	010	000	100	010	110
			Ту	pe II			
Low-	High		-	_			
000	001	010	011	100	101	110	
001	010	011	100	101	110	111	
010	011	100	101	110	111		
011	100	101	110	111			
100	101	110	111				
101	110	111					
110	111						
111							
High	Low						
001	010	011	100	101	110	111	
000	001	010	011	100	101	110	
	000	001	010	011	100	101	
		000	001	010	011	100	
			000	001	010	011	
				000	001	010	
					000	001	
						000	
			Ту	pe III			
			000	111			
			111	000)		
			Ty	pe IV			
	000	001	010 ()11 10	0 101 1	10 111	
	000	001	010 (<u>)11 1</u> 0	0 101 1	10 111	

Delay in	existing	method:
----------	----------	---------

EXISTING METHOD	DELAY
(First category)Bus-Invert	30.77 ns

Delay in proposed schemes:

PROPOSED SCHEME	DELAY
SCHEME 1(half invert)	5.895 ns
SCHEME 2 (half and full inversion)	4.632 ns
SCHEME 3(even and odd inversion)	4.632 ns

In the proposed encoding scheme I, we make use half invert and in scheme II, we make use of both half and full inversion and this scheme compares the current data with the previous one to decide whether half inversion or full inversion of the current data can lead to the link power reduction. Finally in scheme III, we make use even inversion

and odd inversion and this scheme compares the current data with the previous one to decide whether odd inversion or even inversion of the current data can lead to the link power reduction.

Half and Full inversion:

Odd and Even Inversion:



The proposed Encoding and Decoding schemes are simulated and verified using Verilog HDL in Xlinix ISE 10.1i for the target device xc3s500e-5fg320.

V. CONCLUSION

In this paper, we have presented a set of new data encoding and decoding schemes aimed at reducing the power dissipated by the links of an NoC. The self and coupling switching activities are responsible for link power dissipation. In fact, links are responsible for a significant fraction of the overall power dissipated by the communication system. The proposed data encoding and decoding schemes are coded using VERILOG language and is simulated and synthesized using Modelsim and Xilinx software. Overall, the application scheme allows 40% power saving and with less than 5% area overhead in the NI compared to the data encoding scheme.

REFERENCES

- [1] International Technology Roadmap for Semiconductors. (2011) [Online]. Available: http://www.itrs.net
- [2] 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. Mar. 2002, pp. 158–162.
- [3] 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.
- [4] S. Kavitha Student " Data Encoding Technique Using Gray Code in Network-on-Chip ", International Journal Of Science & Technoledge (ISSN2321 – 919X) VLSI Design, Karpagam University, Coimbatore March, 2014.
- [5] A. Sathish, M. Madhavi Latha and K. Lalkishor, "An efficient switching activity reduction technique for network-on-chipdata bus," International Journal of Computer Science Issues, Vol. 8, Issue 4, No2, July 2011.
- [6] 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.
- [7] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," Computer, vol. 35, no. 1, pp. 70–78, Jan. 2002.
- [8] 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.