Data Correlation Aware Serial Encoding for Low Switching Power On-Chip Communication

Somrita Ghosh[†], Prasun Ghosal[†‡], Nabanita Das[*], Saraju P. Mohanty [‡], Oghenekarho Okobiah [‡]
[†]Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, WB, India
[‡]University of North Texas, Denton, TX 76203, USA
[*]Indian Statistical Institute, Kolkata 700108, WB, India
Email: somrita2005@gmail.com, prasun@ieee.org, ndas@isical.ac.in, saraju.mohanty@unt.edu, o0032@unt.edu

***Abstract*—Achieving lightning fast speed data communication in Chip Multi Processor (CMP) based systems as well as Network on Chips (NoCs) is always desired for target performance. Data communication links inside the communication fabric of CMP or NoC architectures have strong impact on their performance and power dissipation. Several approaches exist to reduce power dissipation of parallel link on-chip interconnects, a very few techniques are reported for power reduction in serial links. The existing serial-link power reduction techniques don't necessarily account correlation exhbitted in the data and hence are limited in terms of accuracy. In this paper, a novel data encoding scheme is proposed for serial links to decrease the number of self transitions to reduce the power in data transmission. The proposed scheme accounts the correlations in the data and hence is more effective for real-life applications. The system architecture as well as the encoding and decoding schemes have been implemented to explore the proposed algorithm applicable for any CMP or NoC architectures. The proposed encoding scheme has been analyzed with various types of real-life data streams. Experimental results show that up to 27% reduction in power dissipation is possible in NoC links by the proposed scheme.**

I. Introduction and Motivation

A. Introduction

System design using the nanoscale CMOS (nano-CMOS) technology is dominated by the issues originating from the on-chip communication infrastructures. In multicore architectures, as number of cores in a system-on-chip (SoC) increases, so increases the need of support of proper interconnect solution for communication among the cores. Existing solutions are mainly based on the combinations of shared-buses and dedicated inter-module wires and have scalability issues. Such solutions may not be adequate for nanoscale technologies and multicore architectures. To address this issue, the network-on-chip (NoC) architectures have been developed those are scalable and supportive to current trends of SoC integration in a better way with highly utilized and shared communications. Highly scalable modular structures of NoC are promising replacements for SoC designs with a large number of processing cores [1]–[3]. Besides thermal and reliability issues, two most important design optimization metrics of ULSI design today are energy consumption, and power dissipation. These are important for the lifetime of the battery driven portable mobile systems. Primary constraints from NoC architectures are power dissipation and silicon area manifested in designs consisting of short length interconnects of simple routers and protocols.

As technology shrinks to nanoscale domain, links to routers power ratio increases among communication resources in a NoC and links are becoming more power hungry than routers. This problem is increasing day by day with continuous increasing demand of more faster on chip data transfer with voluminous data used for high performance computing. This phenomenon recently drawn the attention of researchers and efforts are there to reduce power consumption during on chip communication. However, the existing research are primarily focussed on the parallel communication links and the research for the *serial communication links in NoC* is severely lacking.

B. Related Research

Most of the energy in traditional SoCs are dissipated in data buses and long interconnects due to dynamic power consumption during charging and discharging of internal node capacitances as well as inter-wire capacitances [4]. Crosstalk noise is dominated by inter-wire capacitance during charging as well as discharging. A number of encoding techniques have been proposed in the existing literature for the reduction the transitions on the data bus.

Most of the communication in a NoC architecture is through parallel buses. However, the parallel communication suffers from many a disadvantages including routing difficulties, inter-wire spacing, shielding to avoid crosstalk noises, high leakage power due to large driver or receiver, presence of repeaters for long distance communicates, and larger chip area [5], [6].

The serial links in NoC architectures can provide savings in the power dissipation, reduction of wire area, reduction of noise, simpler layout and timing verification, and controlled throughput by adjusting the frequency of the serializer. It also eliminates the requirements of multiple line drivers and buffers [7]. By removing multiple wires serial communication has an additional advantage that it eliminates the skew uncertainty. Disadvantages of serial communication, such as inter symbol interference between successive bits and high speed operation, can be appropriately handled by proper encoding and asynchronous protocols [8].

For parallel data buses, a popular encoding technique is the bus encoding technique [9]. Different types of bus invert cod-

ing schemes include partial bus coding technique [10], and decomposition approach [11]. Other popular encoding techniques include Gray coding, adaptive coding [12] etc. However, the conventional parallel bus encoding techniques cannot be used for serial bus and hence needs research. Transmission energy is minimized by serialized low-energy transmission coding on the serial interconnect by using data correlation properties that might be compromised during serialization [13]. Such serial encoding techniques are data dependent and if correlation range falls within the overhead region, then the results are opposite, i.e power dissipation increase than original data [14]. Moreover, the switching activity factor in serial links is different from that of parallel wires. Sometimes, number of transitions may grow significantly due to serialization [14]. Therefore, it is more crucial to reduce number of transitions independent of data types.

C. Paper organization

Rest of the paper is organized as folllows. Section II highlights the novel contributions of this paper. Proposed methods and algorithms of encoding as well decoding for serial links are described in section III. Section IV discusses details of proposed architectures for encoders as well as decoders. Experimental details, validation, and analysis of results are reported in section V. Finally, section VI concludes the paper with possible future direction of research.

II. Novel Contributions of the Paper

This paper proposes an on-chip data serialization technique with encoding for serial links. The novel contributions of this paper to the state-of-art are the following:

- First, the proposed technique reduces the number of transitions significantly as well as guarantees a low-power dissipation during on chip communication through NoC links.
- Second, proposed method of data encoding/decoding is completely independent of data types and their correlations unlike the shortcomings present in other previous works.
- Third, proposed encoder and decoder parts are designed and implemented using Synopsis Design Vision tool mapped into UMC 90 nm technology node for validation of real operation and results.
- Extensive experiments have been carried out over different types of real data files (with different extensions and sizes) under communication under proposed system architecture to show average power reduction within the range from 20 to 27% for various data file types.
- Comparative discussions with related research results using similar existing techniques.

III. Proposed Encoding and Decoding Methods

A. Problem Formulation and Energy Model

Communication between source and destination nodes takes place by packets forwarding by switches inside a NoC and network interfaces (NI) [15]. Let us consider the average power dissipation per hop (APH) of the NoC. It is the average dynamic power dissipation in a single hop of a packet transmitted over the NoC. APH can be split into three components as follows [16]:

- Average power consumed by a router for switching (APR). The key components of a router are buffers,router wires and controlling logic gates. For better analysis, total router power can be achieved by dividing it into Buffer power consumption (APB) and control components power consumption (APS). Reason for this consideration is that buffers are the largest power consumer for a typical switch or router. Due to this power consumption is much affected by the router buffer than router control due to bit transition effect.
- Average power consumed on a link between two routers (APL).
- Average power consumed on a local link between the router and the system core attached directly to it (APC).

Therefore the average power consumption of a packet transmitted through a router, a local link and a link between routers is given by the following:

$$APH = APR + APL + APC. \tag{1}$$

In regular NoC architectures, the core inputs/outputs are placed near the router local channel. APC is much smaller than APL. APC may be neglected without significant accuracy loss in the estimation of total power dissipation. So, the average router-to-router communication power dissipation is given calculated by the following expression:

$$RRP_{ij} = \eta(APB + APS) + (\eta - 1)APL, \tag{2}$$

where η corresponds to the number of routers through which the packet passes.

The above expression can be rewritten in the context of average power dissipations of the encoder and decoder architectures in the NoC router in the following manner:

$$RRP_{ij} = \eta(APB + APS) + (\eta - 1)APL + (APE + APD), \tag{3}$$

where APE is the average power dissipation of the encoder and V is the average power dissipation of the decoder of the NoC router. Link power primarily depends on the dynamic power consumption. Let us consider a model of n-bit wide bus. C_s is the self or substance capacitance, and C_I is the inter-wire capacitance. The effect of C_I is increasing at a rapid phase leading to increase in the technology with scaling. Thus the effect of both the coupling and substrate activity must be considered in the present technology. The total power dissipated on a bus is given in by the following [17]:

$$P_{bus} = (\alpha_s C_s + \alpha_I C_I)V^2 f, \tag{4}$$

where α_s represents the switching activity of the line and α_I represents the inter-wire switching activity, V represents the supply voltage and f represents the maximum operating clock frequency.

Primary objective of the research is to propose an encoding technique that can minimize the α_s term i.e. self switching activity factor. If we use serial link in place of parallel link

then coupling capacitance between adjacent lines and inter-wire switching activity both will decrease, and power can be reduced to the following [17]:

$$P_{bus} = \alpha_s C_s V^2 f. \quad (5)$$

Thus, the reduction of self switching activity in serial links will lead to the reduction in dynamic power consumption.

B. Proposed Encoding Technique

In serial data stream, proposed encoder checks every consecutive strings of length three of each byte, if it contains two consecutive transitions like $0 \rightarrow 1 \rightarrow 0$ or $1 \rightarrow 0 \rightarrow 1$ then last two bits are swapped. For example,

$$\begin{array}{ccc} 010 & \rightarrow & 001 \\ 101 & \rightarrow & 110 \end{array}$$

Other bits remain unchanged. For each byte, $(a_0 a_1 ... a_7)$ two additional lines L_1 and L_2 are introduced to keep track of the changes in bits by encoding. If major part of the three-bit string changed lies within $(a_0 a_1 a_2 a_3 a_4)$, (or in $(a_3 a_4 a_5 a_6 a_7)$) then $L_1 = 1$ and $L_2 = 0$ (or $L_1 = 0$ and $L_2 = 1$) respectively. For no change $L_1 = L_2 = 0$, and similarly for changes in both halves $L_1 = L_2 = 1$.

Table I shows how this proposed method works on data-streams and how much reduction of transition can be achieved. Algorithm 1 outlines proposed encoding technique.

Proposed_Encoding_Technique( )

Input : Dataset $(A = a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$
Output: Encoded Dataset$(B = b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7)$, L_1, $L_2)$

```
begin
    Length = Length(A);
    i = 0;
    L1 = 0;
    L2 = 0;
    B = A;
    for i ≤ Length − 3 do
        if a_i = a_{i+2} ≠ a_{i+1} then
            b_{i+2} = a_{i+1};
            b_{i+1} = a_{i+2};
            if i < 3 then
                L2 = 1;
            else
                L1 = 1;
            end
            i = i + 2;
        end
    end
end
```

Algorithm 1: Algorithm for Proposed Encoding Technique to Reduce the Number of Transitions.

C. Proposed Decoding Technique

Pseudo code for our proposed decoding technique is presented in Algorithm 2. In this algorithm according to the contents of additional lines it will swap the corresponding bits and regenerate actual dataset.

Proposed_Decoding_Technique( )

Input : Encoded Dataset$(B = b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7)$, L_1, $L_2)$
Output: Decoded Dataset $(A = a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$

```
begin
    A = B;
    if L2 = 1 then
        i = 0;
        for i ≤ 2 do
            if b_i = b_{i+1} ≠ b_{i+2} then
                a_{i+1} = b_{i+2};
                a_{i+2} = b_{i+1};
            end
        end
    end
    if L1 = 1 then
        i = 3;
        for i ≤ 5 do
            if b_i = b_{i+1} ≠ b_{i+2} then
                a_{i+1} = b_{i+2};
                a_{i+2} = b_{i+1};
            end
        end
    end
end
```

Algorithm 2: Algorithm for Proposed Decoding Technique to Generate Actual Dataset.

D. Correctness Proof

Let us take an arbitrary data stream of eight bits ($a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7$ = 11010101) and want to perform the proposed encoding technique on them. According to the proposed encoding method three consecutive bits are checked at a time. In the above data stream, a_1, a_2, a_3 have two consecutive transitions between them - first $1 \rightarrow 0$ and second $0 \rightarrow 1$. Therefore in encoded stream a_2, a_3 will exchange their positions and 101 will become 110. For this string index i = 1 which will make additional line $L_2 = 1$. After this, next three bits(a_4, a_5, a_6=010) are checked and as they are also satisfying the condition they are changed into 001. At this position index i=4 and this will make $L_1 = 1$. So the final encoded data stream will look like 11100011 and additional line contents will be $L_2 L_1 = 11$.

At the decoder part, additional lines are checked. As described in decoding algorithm 2 among the received encoded data b_1, b_2, b_3(=110) and b_4, b_5, b_6 (=001) satisfies the condition of swap. Therefore b_2, b_3 and b_5, b_6 are swapped. Finally, decoded data stream changes to 11010101 which is the original data.

Theorem 1: Total number of transitions will never increase for any combination of dataset after applying the proposed encoding technique.

Proof: We can see that in our proposed method if the condition is satisfied then a_{i+1} and a_{i+2} will swap their positions.It will decrease one transition for sure. Now after this swapping a_{i+2} is flipped and a_{i+3} remains the same, even if it is a part of a changed sequence (since it is the first bit). Hence two situations may arise as follows:

1) a_{i+2} and a_{i+3} are same (originally they different).
2) a_{i+2} and a_{i+3} are opposite (originally they were same).

In the first case, two transitions are saved. Whereas in the second case, an additional transition is introduced between

TABLE I
AN EXAMPLE OF PROPOSED ENCODING TECHNIQUE.

Before Encoding		After Encoding		Additional	Reduction
Datastream	No. of Transitions	Datastream	No. of Transitions	Lines(L_2,L_1)	of Transitions
1 **101** **010** 1	6	1 **110** **001** 1	2	11	4
11 **101** 001	4	11 **110** 001	2	10	2
00110 **010**	4	00110 **001**	3	01	1
11110001	3	11110001	3	00	0

a_{i+2} and a_{i+3}, which will nullify the transition reduction gain by swapping. Finally we conclude that the total number of transitions remains either same or reduced after encoding. It proves the theorem.

E. Data Correlation Independence

To justify that the proposed technique achieves power reduction independent of data correlation, we defined a set of test-scenarios for various traffic patterns. This set contains three levels as follows:

- Best case (no bit-flip).
- Worst case (bit-flips in both halves).
- Typical case (random data with 50% bit-flips, or bit-flips in one half only).

All possible test cases with corresponding number of transitions are shown in Table II that supports the claim that the reduction in power is independent of data types.

Fig. 1 shows the variation of reduction in number of transitions for different bus widths. It shows that on an average the reduction in total number of transitions for different bus widths is near about 25%.

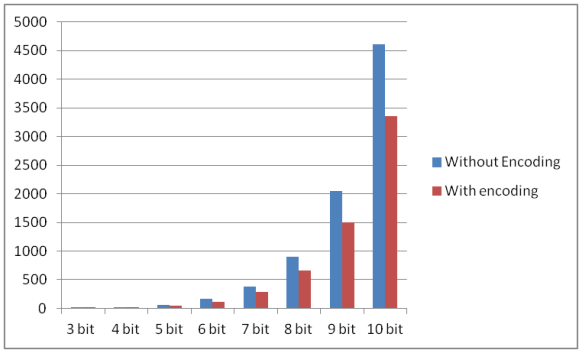

Fig. 1. Total Number of Transitions for Different Bus Widths with or without Encoding.

IV. PROPOSED ENCODER AND DECODER ARCHITECTURES

Encoder and decoder structures are placed within the network interface module of the proposed architecture. This placement reduces power dissipation in the links and also eliminates the necessity for buffers and additional structures to provide a low-power datapath.

A. Placement of Encoder and Decoder

Serializer, deserializer, and the proposed *CODEC* circuits are integrated in each network interface (NI). In the proposed architecture shown in Fig. 2, encoder, decoder, serializer, and deserializer are placed at the NI level. Let us assume that M bit data is sent over N parallel lines from the core. In SILENT method [13], first parallel data has been encoded and then that is serialized. Therefore more number of encoders are needed to implement this technique, which in turn increases chip area and power consumed. But in proposed architecture, encoder structure is placed following the data serializer and each encoder operates on the serialized data. So we require only one encoder and decoder structure for each NI. Each encoder and decoder works on 8 bit data. Encoded data goes through the serial dataline from switch A to switch B along with additional lines as shown in Fig.2. Though introduction of two extra lines have a little area and power overhead, but there no further degradation of speed of operation is noticed.

B. Design and Synthesis of Encoder and Decoder

The schemantic encoder-decoder is realised using Synopsys Design Vision tool, mapped into a UMC 90 nm technology. Serialization and de-serialization are done here by multiplexer serialization and de-serialization technique. Details are omitted due to paucity of space.

The encoder design consists of three sections as follows:

1) first to check serial input data and compare with the condition.
2) if condition is satisfied then data is swapped.
3) generate proper control bits for additional lines.

Principle of decoder design consists of two different steps as follows:

1) to check the additional lines.
2) swapping the received data bits according to the additional line.

V. EXPERIMENTAL RESULTS AND VALIDATION

A. Comparison of Proposed Technique with SILENT Encoding

Proposed encoding technique is compared with SILENT coding for an arbitrary data-streams with 8 parallel data lines (A0 to A7) and each line having 8 bits of data [13]. Table III shows a 22.58% reduction by proposed method over SILENT encoding with 12.9% reduction only. Fig. 3 shows the reduction of transition by SILENT coding and the proposed coding.

TABLE II
AN EXAMPLE OF PROPOSED ENCODING TECHNIQUE WITH ALL POSSIBLE TEST SETS.

Type of case	Before Encoding		After Encoding		Additional
	Datastream	No. of Transitions	Datastream	No. of Transitions	Lines(L_2,L_1)
Best Case	11111111	0	11111111	0	00
Worst Case	10101010	7	11000110	3	11
Typical Case	01010001	5	00110001	3	10
Typical Case	11001010	5	11000110	3	01

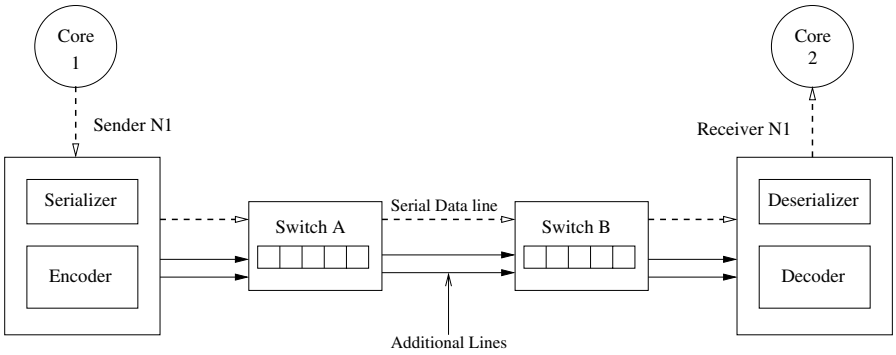

Fig. 2. Proposed Architecture of the System.

TABLE III
COMPARISON OF SILENT ENCODING AND PROPOSED ENCODING ON AN ARBITRARY DATASTREAM.

Parallel Lines	Data stream	# Trans w/o Enc	SILENT [13] Enc	# Trans SILENT [13]	Proposed Enc	# Trans w Enc
A0	11010101	6	10111111	2	11110001	2
A1	11011010	5	10110111	4	11100110	3
A2	10111011	4	11100110	3	00111100	2
A3	01100000	2	01010000	4	11110110	3
A4	10110111	4	11101100	3	01100110	4
A5	01111110	2	01000001	3	10001110	3
A6	11001001	4	10101101	6	01110100	4
A7	00010100	4	00011110	2	11000110	3
		Total=31		Total=27		Total=24

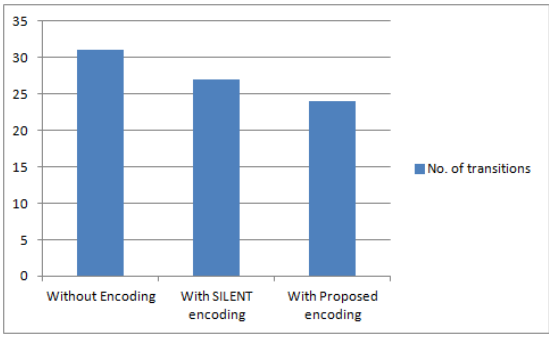

Fig. 3. Reduction of Transitions with respect to SILENT Encoding.

B. Experimental Set Up

Proposed algorithm was modeled in VHDL and synthesized with Synopsis Design Compiler and Prime time tools and mapped onto a UMC 90 nm technology library. In worst case, area requirement for encoder is $681.169nm^2$ and for decoder it is $471.944nm^2$ as shown in Table IV. This encoder was tested against different types and sizes of data streams. It includes jpg file, png file, PDF file, text, doc file, HTML file, and MP3 file.

TABLE IV
AREA AND POWER REQUIREMENT FOR DESIGNED ENCODER AND DECODER

	Encoder	Decoder
Power	$0.172mW$	$0.112mW$
Area	$681.169nm^2$	$471.944nm^2$

TABLE V
REDUCTION IN NUMBER OF TRANSITIONS FOR DIFFERENT TYPES OF FILES WITH PROPOSED ENCODING TECHNIQUE.

File Type	File size	% Change by SILENT [13]	% Change by Proposed Encoding
jpg1	5.41KB	-1.78	-26.77
jpg2	27.9KB	-7.89	-26.66
jpg3	81.8KB	**+3.19**	-26.73
jpg4	103.1KB	**+6.26**	-25.40
png1	1.25MB	**+0.06**	-26.35
png2	58.5KB	-0.93	-26.40
png3	683.3KB	-0.66	-26.08
png4	516.8KB	**+0.50**	-26.79
pdf1	95KB	-0.47	-24.95
pdf2	275.3KB	-0.12	-26.30
pdf3	694.4KB	**+0.09**	-23.59
pdf4	1.3MB	**+0.28**	-26.59
text1	18KB	-2.84	-22.47
text2	4.4KB	-7.92	-20.13
text3	33.1KB	-1.82	-22.67
text4	64.7KB	-1.05	-22.54
Html	73.5KB	**+6.62**	-23.28
MP3_1	378.5KB	-0.57	-27.82
MP3_2	1.9MB	**+0.54**	-27.44
MP3_3	6.8MB	-0.38	-27.13
MP3_Blind	2.6MB	**+4.72**	-27.87

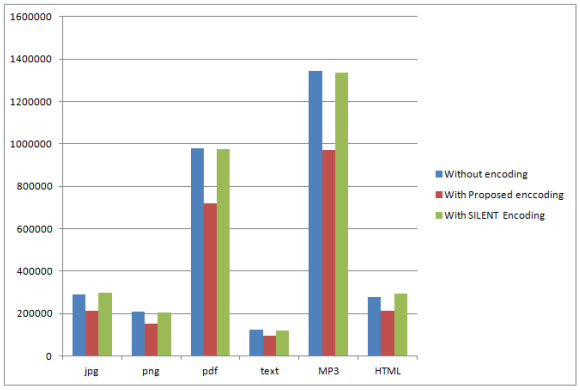

Fig. 4. Number of Transitions with and without Encoding and SILENT Coding for Different Types of Files.

C. Discussions

It is evident from experimental results that proposed scheme obtains on an average 26.39% reduction in number of transitions in case of JPG file, 26.406% for PNG file, 25.36% for PDF file, 21.95% for TEXT and 27.56% for AUDIO (MP3) file and 23.27% for HTML file (Table V), +ve sign indicates the increase in transition and -ve sign indicates the decrease in transition.

In comparison to SILENT coding proposed technique is much more effective where data bits are highly correlated and power reduction is not always positive as shown in Table V. Unlike SILENT coding, proposed method always shows improvement in case of switching activity and thereby power reduction. Fig. 4 plots the number of transitions without encoding and with encoding and SILENT coding for different types of files. SILENT coding shows the best performance for instruction address, about 77% energy saving for 3D graphics but for random traffic it can achieve only 13% of energy savings [14]. Proposed method obtains above 20% of energy savings in all the cases and guarantees that it will either reduce the number of transitions or keeps it the same.

The bus should be as long as 15 mm at least to hide the power overhead of CODEC circuits to save 20% power on a parallel bus by using parallel bus encoding techniques such as Bus-Invert coding [9], [18]. The length to obtain significant power savings is even longer. Situation is worse in nanoscale technology. The proposed coding is designed for a serial bus and ensures power reduction irrespective of data types without need for long length bus.

VI. Conclusion and Future Work

This paper presents a novel encoding technique to minimize the number of transitions in the serial link of NoC router to reduce dynamic power consumption due to switching activity in the capacitances. Proposed method has been compared with SILENT encoding technique, which, is only effective for multimedia data applications. Proposed encoding structure has been implemented that always leads to significant reduction in switching activity. Proposed framework is tested for different types of data streams while accounting the data correlations. It was observed that unlike the SILENT coding, power savings is achieved for all data types. Average reduction is near about 25%, which, is far better than SILENT coding. However, addition of two extra lines causes a little area and power overhead and it requires special attention. In this paper, only self switching activity of the links was considered. Future research will focus on minimizing area and power overhead by suitable techniques, making it more robust by considering other effects like inter symbol interference, also to integrate the benefit considering other dynamic power dissipation factors.

References

[1] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *Computer, IEEE*, vol. 35, no. 1, pp. 70–78, 2002.

[2] A. Jerraya and W. Wolf, *Multiprocessor Systems-on-chips*. Elsevier, 2004.

[3] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-chip," *ACM Computing Surveys (CSUR)*, vol. 38, no. 1, p. 1, 2006.

[4] S. P. Mohanty, N. Ranganathan, and S. K. Chappidi, "ILP Models for Energy and Transient Power Minimization During Behavioral Synthesis," in *Proceedings of the 17th International Conference on VLSI Design*, pp. 745–748, 2004.

[5] M. Pedram and J. M. Rabaey, *Power Aware Design Methodologies*. Springer, 2002.

[6] T. Sakurai, "Perspectives on Power-aware Electronics," in *Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. 2003 IEEE International*, pp. 26–29, IEEE, 2003.

[7] D. N. Sarma and G. Lakshminarayanan, "Encoding Technique for Reducing Power Dissipation in Network on Chip Serial Links," in *Proceedings of the International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 323–327, 2011.

[8] A. Morgenshtein, I. Cidon, A. Kolodny, and R. Ginosar, "Comparative Analysis of Serial vs Parallel Links in NoC," in *System-on-Chip, 2004. Proceedings. 2004 International Symposium on*, pp. 185–188, IEEE, 2004.

[9] M. R. Stan and W. P. Burleson, "Bus-invert Coding for Low-power I/O," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 3, no. 1, pp. 49–58, 1995.

[10] Y. Shin, S.-I. Chae, and K. Choi, "Partial Bus-invert Coding for Power Optimization of Application-specific Systems," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 377–383, 2001.

[11] S. Hong, U. Narayanan, K.-S. Chung, and T. Kim, "Bus-invert Coding for Low-power I/O - A Decomposition Approach," in *Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium on*, vol. 2, pp. 750–753, IEEE, 2000.

[12] L. Benini, A. Macii, M. Poncino, and R. Scarsi, "Architectures and Synthesis Algorithms for Power-efficient Bus Interfaces," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 19, no. 9, pp. 969–980, 2000.

[13] K. Lee, S.-J. Lee, and H.-J. Yoo, "SILENT: Serialized Low Energy Transmission Coding for On-chip Interconnection Networks," in *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pp. 448–451, IEEE Computer Society, 2004.

[14] K. Lee, S.-J. Lee, and H.-J. Yoo, "Low-power Network-on-chip for High-performance SoC Design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 148–160, 2006.

[15] F. Gebali, H. Elmiligi, and M. W. El-Kharashi, *Networks-on-chips: Theory and Practice*. Taylor & Francis, US, 2011.

[16] J. Palma, L. Indrusiak, F. Moraes, A. Garcia Ortiz, M. Glesner, and R. Reis, "Inserting Data Encoding Techniques into NoC-Based Systems," in *VLSI, 2007. ISVLSI '07. IEEE Computer Society Annual Symposium on*, pp. 299–304, 2007.

[17] K. Sainarayanan, J. Ravindra, and M. Srinivas, "A Novel, Coupling Driven, Low Power Bus Coding Technique for Minimizing Capacitive Crosstalk in VLSI Interconnects," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pp. 4–pp, IEEE, 2006.

[18] C. Kretzschmar, A. K. Nieuwland, and D. Muller, "Why Transition Coding for Power Minimization of On-chip Buses Does Not Work," in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, vol. 1, pp. 512–517, IEEE, 2004.