

Adaptive Word Reordering for Low-Power Inter-Chip Communication

Eleni Maragkoudaki

Advanced Processor Technologies Group
School of Computer Science
University of Manchester
eleni.maragkoudaki@manchester.ac.uk

Przemyslaw Mroszczyk

Qualcomm
Cork, Ireland
przemyslaw.mroszczyk.23@gmail.com

Vasilis F. Pavlidis

Advanced Processor Technologies Group
School of Computer Science
University of Manchester
pavlidis@cs.man.ac.uk

Abstract—The energy for data transfer has an increasing effect on the total system energy as technology scales, often overtaking computation energy. To reduce the power of inter-chip interconnects, an adaptive encoding scheme called Adaptive Word Reordering (AWR) is proposed that effectively decreases the number of signal transitions, leading to a significant power reduction. AWR outperforms other adaptive encoding schemes in terms of decrease in transitions, yielding up to 73% reduction in switching. Furthermore, complex bit transition computations are represented as delays in the time domain to limit the power overhead due to encoding. The saved power outweighs the overhead beyond a moderate wire length where the I/O voltage is assumed equal to the core voltage. For a typical I/O voltage, the decrease in power is significant reaching 23% at just 1 mm.

Index Terms—Interconnects, Inter-chip communication, Encoding, Low power, Adaptive word reordering

I. INTRODUCTION

The ever growing compute density enabled by parallelism constantly increases the demand for data transfer [1]. In addition, the energy of transferring data, especially off-chip, increases, surpassing the energy for computation [2]. Thus, the power reduction of inter-chip communication is one of the key challenges for modern integrated systems.

An efficient way to reduce the power of interconnects is to decrease signal transitions through data coding. Encoding schemes can be classified as either static or adaptive. Static schemes ([3]-[8]) exploit the statistical properties of the data stream, which are considered known at design time. However, having *a priori* knowledge of data statistics is not always feasible or the characteristics of the data may vary over time.

Adaptive schemes observe the data stream during system operation and apply appropriate encoding. Bus Invert (BI) [9] is one of the early encoding schemes, which inverts the data word if more than half of the bits switch. The Adaptive Partial Bus Invert (APBI) is an extension of BI, which periodically forms a subgroup of encoded lines [10]. The working zone is tailored to address buses and assumes that applications use specific subgroups of the address space [11]. Frequent Value is an adaptive technique that encodes the frequently transmitted words [12]. These words are stored

This work was supported in part by the European Commission under the Horizon 2020 Framework Programme for Research and Innovation through the EuroExa project (grant agreement 754337).

in a Content-Addressable Memory (CAM), which increases the power and hardware requirements. Adaptive dictionary encoding [13] uses a dictionary to store recurring patterns in order to reduce the number of bits. Adaptive Bus Encoding (ABE) [14] observes the data stream, thereby forming clusters with the highly correlated lines of the bus.

All these adaptive schemes either do not provide high transition reduction or have significant power overhead and, hence, allow savings for extremely large capacitive loads. To address these drawbacks, an adaptive encoding scheme, namely Adaptive Word Reordering (AWR), is proposed for wide off-chip buses. The new scheme provides high switching reduction for diverse data types without affecting the communication throughput. The encoder and decoder circuits exhibit a low overhead in power, thus, power can be saved even for short interconnect lengths.

The rest of this paper is structured as follows. The proposed technique and the circuit implementation are described in Sections II and III, respectively. The power gains are quantified in Section IV and a comparison with existing encoding schemes is provided. Finally, conclusions are drawn in Section V.

II. THE ADAPTIVE WORD REORDERING METHOD

The proposed encoding scheme is based on the observation of the data stream over fixed windows of N words and the dynamic reordering of these words to decrease the number of transitions. The problem of the optimal word reordering is equivalent to the Travelling Salesman Problem (TSP) [15], classified as NP-hard. Each word is considered as a vertex of a complete, undirected graph $G(V, E)$. The weight of each edge, w , is the Hamming distance of each pair of words. The problem is reformulated as the identification of the minimum weight route that visits all the vertices of the graph exactly once. Although the route in TSP is cyclic and has to end to the starting vertex, the computational complexity is the same.

The hardware complexity of exact algorithms is prohibitive if any savings in power are to be harvested. Therefore, the Nearest Neighbour (NN) search is implemented, as a less complex heuristic approach. The NN algorithm is properly adapted for the specific problem, where the search starts at the previously transmitted word and each time visits the vertex with the minimum weight that has not been visited yet.

III. AWR CIRCUIT AND INTERCONNECT MODEL

The hardware architecture of the proposed encoding scheme is provided in this section. The encoder and decoder circuits are described in subsection III-A and the model of the interconnect used for the simulations is presented in subsection III-B. The circuits are designed with a 65 nm technology [16].

A. Encoder and Decoder Circuits

The encoding of data is implemented as follows. In each clock cycle, the Hamming distances between the previous word and the words that have not yet been transmitted are evaluated and the word with the lowest Hamming distance is transmitted over the bus. Conventionally, the computation of the Hamming distance is implemented using adder trees. This approach is highly inefficient in terms of power, hence, a different approach is followed. The Hamming distance is determined as a delay in the time domain, drastically reducing the power for encoding.

This method requires N registers at the transmitter and the receiver to store the reordered words and the maximum overhead in latency is $2N$ clock cycles. However, encoding does not limit the communication bandwidth, as a data transfer takes place over each clock cycle.

The proposed encoder circuit comprises three stages, as depicted in Fig. 1. The first stage is the race stage, where a variable delay line is assigned to each word. In each delay line, a clock pulse is propagated and is delayed according to the number of bits that switch. The delay is shorter for a lower number of transitions and, thus, the fastest signal corresponds to the word with the lowest Hamming distance. A detailed description of the delay line can be found in [17].

The signal that arrives first to the finish stage prevents the others from propagating. Before any signal arrives, the value 0 is stored in all the latches (LA) and the PER signal is set to 1 using a weak pull-up resistor. The signal that arrives first, sets the respective latch and resets PER. The buffers (BUF) add a short delay to the DEL signal to ensure that “1” is stored in the latch before the latch is disabled.

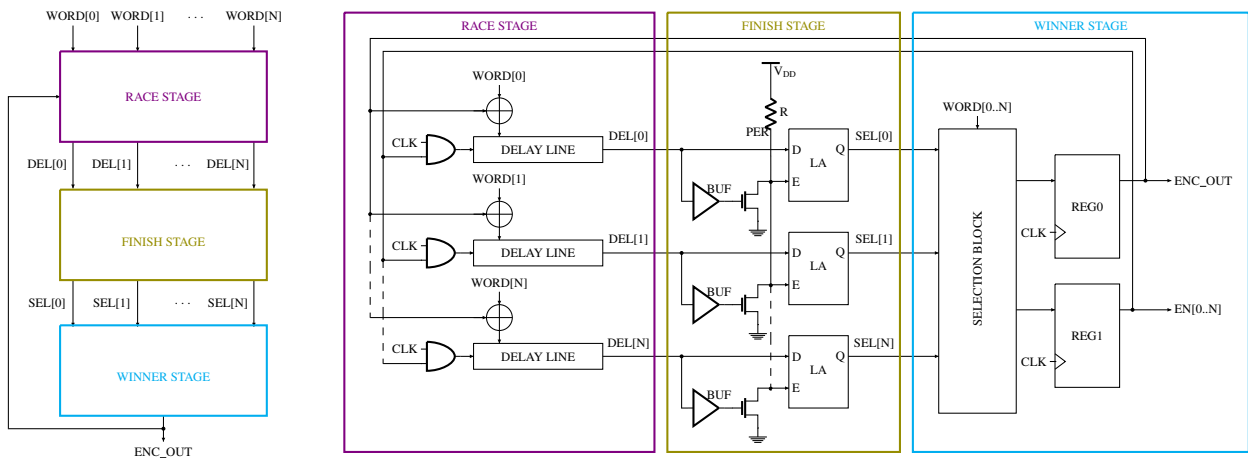


Fig. 1: Encoder circuit.

The winner stage is composed by the selection block and two registers. The selection block is a digital circuit that decides which word wins the race according to the received $SEL[0..N]$ signals. In case two or more signals arrive at the same time, the word with the lowest index is chosen. The winning word is stored in the register REG0. In REG1 the $EN[0..N]$ signals are stored, which are used to keep track of the transmitted words.

To guarantee that the data can be placed back in the initial order, low spatial redundancy is used. K more bits are added to each word to indicate the order. Thus, for N words, $K = \log_2 N$ additional bus lines are required. These bits are included in the encoding to ensure that they do not considerably increase the power overhead. A K -to- N decoder is used to store the words to the registers in the right order.

B. Interconnect Model

The effectiveness of the proposed AWR method is explored for an inter-chip link for 2.5-D integration as interposers support a high wire density. The link is assumed to connect two dies that are bump bonded on top of a silicon-based interposer. The circuit of the interconnect consists of the distributed wire model, I/O cells, and the parasitic capacitance of μ bumps, $C_{\mu bump} = 30$ fF [18]. For the sake of simplicity, the mutual inductances are not considered. The global wire dimensions for a 65 nm technology are used according to [19]. The electrical characteristics of the wires for minimum pitch are listed in Table I.

IV. RESULTS

The efficiency of the proposed encoding scheme is validated in terms of the decrease in both transitions and power in subsections IV-A and IV-B, respectively. To determine the

TABLE I: Electrical characteristics of wires.

R [Ω/mm]	L [nH/mm]	C_{GND} [fF/mm]	C_C [fF/mm]
40.74	1.52	222.55	52.45

theoretical decrease in transitions, the encoding schemes are simulated using Matlab. The power of the inter-chip link and the related circuits is measured using Spectre, Cadence® at nominal conditions (typical device corners, 27°C).

A. Switching Reduction

The theoretical performance of the proposed reordering technique is reported in Table II and is compared with three other adaptive encoding schemes, BI [9], APBI [10], and ABE [14]. The first five benchmarks comprise a mix of memory addresses and data and are generated from different applications. LFRic [20] is a weather forecasting and climate research atmospheric model, InFoli is a simulator of the brain Inferior Olive, and LITE_LOOP, LU_SOLVER, and NASTY_EXPS are microkernels from the Integrated Forecast System [21]. Furthermore, one black and white and one coloured image are used, as well as uniformly distributed random data generated with Matlab. For these simulations, an observation window of $N = 32$ words is used, while the bus width is $M = 64$ bits.

The AWR scheme provides greater self-switching savings than the other methods for all types of data streams. A higher decrease in switching is achieved for the image files reaching up to $\sim 73\%$. Furthermore, AWR is highly effective for multiplexed address-data buses as the provided reduction is about double compared to prior art. AWR is also useful in physical media where the coupling capacitance is significant as the decrease in relative transitions of adjacent lines is remarkable. On the contrary, other techniques are not consistent as they can increase the number of relative transitions in some cases.

B. Power Gains

The power savings of AWR are validated for two scenarios using the LFRic benchmark. First, the typical I/O voltage and core voltage are considered for the 65 nm technology, $V_{DDIO} = 1.8$ V and $V_{DDCORE} = 1.2$ V. The operating frequency is 400 MHz and the bus width is $M = 64$ bits. The efficiency of AWR is estimated with respect to the interconnect length for different number of reordered words, N . The power overhead ranges from 2.5 to 6.3 mW and the critical path delay is 2.1 ns for $N = 32$ which is comparable to the interconnect delay. As illustrated in Fig. 2, the power gains are remarkable, reaching 23% at just 1 mm length. The highest reduction is achieved for $N = 64$ because of the greater switching reduction, even though the overhead in power is higher.

The limitations of the proposed technique are explored in a second scenario with a lower I/O voltage ($V_{DDIO} = 1.2$ V). The I/O cells are removed and an ultra low capacitance for the electrostatic discharge protection of the I/O cells is added ($C_{ESD} = 115$ fF [18]). The decrease in power is depicted in Fig. 3 for a different number of reordered words, while the bus width is fixed to 64 bits. As expected, for short interconnects < 6 mm, AWR is not effective as the power consumed in the interconnect decreases, while the power overhead is constant. In this case, $N = 32$ words is more efficient because of the lower power overhead. Furthermore, in Fig. 4 is shown that the efficiency of the proposed method is higher for wider buses. The highest reduction in power is achieved for $M = 128$ bits, reaching 10% reduction at 11 mm. However, in this case the frequency is reduced to 200 MHz due to the longer delay lines in the encoder circuit. Note that the power efficiency can increase if a more advanced technology process is used.

AWR is compared with the BI scheme, which exhibits the lowest power overhead out of all the adaptive schemes. The first scenario with $V_{DDIO} = 1.8$ V is considered, with $N = 32$ and $M = 64$. The frequency is 400 MHz and the wire length is 1 mm. The encoder and decoder of BI are described in VHDL and synthesized using Design Compiler, Synopsys. In Table III, the overhead as well as the power consumed in the interconnect are reported for the two schemes. AWR provides significantly higher power savings for all the image files and the multiplexed address and data benchmarks, demonstrating that the power overhead of AWR is greatly outweighed in cases with high reduction in switching even for short interconnects. Finally, the benefits of data encoding diminish for random data as AWR just “breaks even” while BI provides $\sim 5\%$ decrease in power.

V. CONCLUSIONS

An adaptive encoding scheme (AWR) is proposed for inter-chip communication which decreases the interconnect power without *a priori* knowledge of the data statistics. AWR changes the order of the words to effectively reduce the switching activity. The encoder circuit exploits the time domain to alleviate the cost of complex computations. The theoretical results demonstrate that AWR outperforms existing adaptive schemes for both data and address-data buses. The power results show that for the typical values of V_{DDIO} and

TABLE II: Decrease in both self and relative switching activity for different encoding schemes and diverse data streams.

Data Streams	Data type	Proposed AWR		BI [9]		APBI [10] ¹		ABE [14] ¹	
		Self	Relative	Self	Relative	Self	Relative	Self	Relative
LFRic	Address-Data	36.88%	10.89%	14.37%	0.57%	16.27%	-12.83%	18.50%	-5.73%
InFoli	Address-Data	35.99%	10.55%	14.92%	0.54%	15.72%	-13.09%	18.12%	-5.74%
LITE_LOOP	Address-Data	36.02%	10.59%	14.91%	0.52%	15.73%	-13%	18.13%	-5.66%
LU_SOLVER	Address-Data	35.91%	10.61%	14.95%	0.53%	15.65%	-13.01%	18.08%	-5.62%
NASTY_EXPS	Address-Data	36.01%	10.58%	14.94%	0.52%	15.75%	-13%	18.12%	-5.69%
BW image	Data	37.28%	39.04%	10.03%	0.18%	15.93%	34.31%	32.56%	41.67%
Colour image	Data	72.99%	74.54%	22.58%	22.56%	37.88%	40.96%	36.02%	36.02%
Random	Data	13.08%	13.03%	8.54%	8.50%	10.59%	11.02%	6.17%	7.53%

¹ Minus sign implies an increase in the number of transitions.

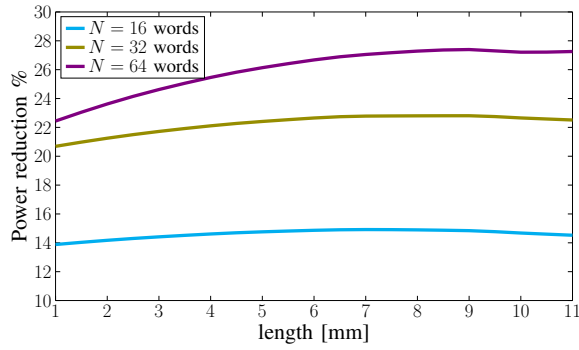


Fig. 2: Decrease in power when $V_{DDIO} = 1.8$ V for $M = 64$ bits and different number of reordered words N .

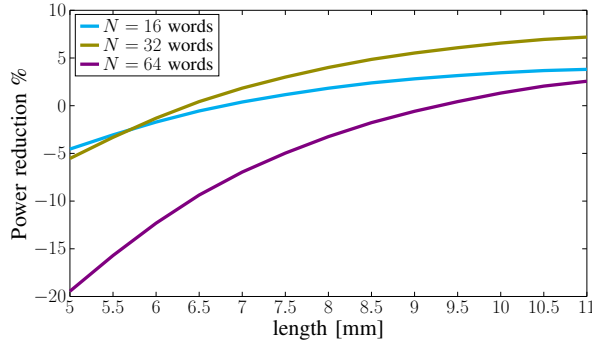


Fig. 3: Decrease in power for $M = 64$ bits and different number of words N ($V_{DDCORE} = V_{DDIO} = 1.2$ V).

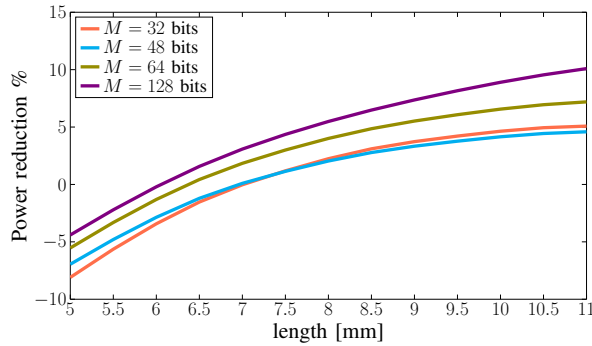


Fig. 4: Decrease in power for fixed number of words $N = 32$ and different bus widths M ($V_{DDCORE} = V_{DDIO} = 1.2$ V).

V_{DDCORE} of the used technology, the decrease in power is remarkable and reaches 23% for address-data buses for just 1 mm interconnects. Finally, the power of the AWR circuit is greatly outweighed by the power saved in the interconnect for data streams with a considerable decrease in switching.

REFERENCES

[1] S. Borkar, "Role of Interconnects in the Future of Computing," *Journal of Lightwave Tech.*, Vol. 31, No. 24, pp. 3927–3933, Dec. 2013.
 [2] P. Kogge and J. Shalf, "Exascale Computing Trends: Adjusting to the "New Normal" for Computer Architecture," *Computing in Science and Engineering*, Vol. 15, No. 6, pp. 16–26, Nov.–Dec. 2013.

TABLE III: Comparison of reduction in power. Power is listed in mW.

Data Streams	Interconnect Power			Overhead		Savings	
	Unencoded	AWR	BI	AWR	BI	AWR	BI
LFRic	39.41	27.49	34.61	3.77	1.53	20.68%	8.31%
InFoli	32.40	21.78	29.93	3.30	1.29	22.59%	3.68%
LITE_LOOP	32.41	21.81	29.94	3.30	1.29	22.52%	3.67%
LU_SOLVER	32.41	21.81	29.94	3.30	1.29	22.52%	3.67%
NASTY_EXPS	32.41	21.81	29.94	3.30	1.29	22.52%	3.67%
BW image	12.02	8.137	11.25	2.96	0.57	7.70%	1.77%
Colour image	55.70	18.33	38.85	3.38	1.78	61.01%	27.05%
Random	58.26	52.16	53.42	5.79	2.19	0.54%	4.55%

[3] C. L. Su, C. Y. Tsui, and A. M. Despain, "Saving Power in the Control Path of Embedded Processors," *IEEE Design and Test of Computers*, Vol. 11, No. 4, pp. 24–30, 1994.
 [4] L. Benini *et al.*, "Asymptotic Zero-Transition Activity Encoding for Address Buses in Low-Power Microprocessor-Based Systems," *Proc. of Great Lakes Symposium on VLSI*, pp. 77–82, March 1997.
 [5] L. Benini *et al.*, "System-Level Power Optimization of Special Purpose Applications: The Beach Solution," *Proc. of the Int. Symp. on Low Power Electronics and Design*, pp. 24–29, Aug. 1997.
 [6] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "A Coding Framework for Low-Power Address and Data Buses," *IEEE Trans. on Very Large Scale of Integr. (VLSI) Syst.*, Vol. 7, No. 2, pp. 212–221, June 1999.
 [7] L. Benini *et al.*, "Architectures and Synthesis Algorithms for Power-Efficient Bus Interfaces," *IEEE Trans. on Computer-Aided Design of Integr. Circuits and Syst.*, Vol. 19, No. 9, pp. 969–980, Sep. 2000.
 [8] Y. Shin, S. Chae, and K. Choi, "Partial Bus-Invert Coding for Power Optimization of System Level Bus," *Proc. of the Int. Symp. on Low Power Electronics and Design*, pp. 127–129, Aug. 1998.
 [9] M. R. Stan and W. P. Burleson, "Bus-Invert Coding for Low-Power I/O," *IEEE Trans. on Very Large Scale of Integr. (VLSI) Syst.*, Vol. 3, No. 1, pp. 49–58, March 1995.
 [10] C. Kretzschmar, R. Siegmund, and D. Mueller, "Adaptive Bus Encoding Technique for Switching Activity Reduced Data Transfer over Wide System Buses," *Proc. of the Inter. Workshop on Power and Timing Modeling, Optimization and Simulation*, pp. 66–75, September 2000.
 [11] E. Musoll, T. Lang, and J. Cortadella, "Working-Zone Encoding for Reducing the Energy in Microprocessor Address Buses," *IEEE Trans. on Very Large Scale Integr. (VLSI) Syst.*, Vol. 6, No. 4, pp. 568–572, December 1998.
 [12] J. Yang and R. Gupta, "FV Encoding for Low-Power Data I/O," *Proc. of the Int. Symp. on Low Power Elec. and Design*, pp. 84–87, Aug. 2001.
 [13] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, "An Adaptive Dictionary Encoding Scheme for SOC Data Buses," *Proc. of the Conf. on Design, Automation and Test in Europe*, pp. 1059–1064, March 2002.
 [14] S. Sarkar, A. Biswas, A. S. Dhar, and R. M. Rao, "Adaptive Bus Encoding for Transition Reduction on Off-Chip Buses With Dynamically Varying Switching Characteristics," *IEEE Trans. on Very Large Scale Integr. (VLSI) Syst.*, Vol. 25, No. 11, pp. 3057–3066, November 2017.
 [15] G. Laporte, "The Traveling Salesman Problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, No. 2, pp. 231–248, 1992.
 [16] United Microelectronics Corporation UMC, <http://umc.com/>.
 [17] M. Fujino and V. G. Moshnyaga, "An Efficient Hamming Distance Comparator for Low-Power Applications," *Proc. of the IEEE Int. Conf. on Electronics, Circuits and Syst.*, pp. 641–644, Sept. 2002.
 [18] P. Mrozczyk and V. F. Pavlidis, "Ultra-Low Swing CMOS Transceiver for 2.5-D Integrated Systems," *Proc. of the IEEE Intern. Symp. on Quality Electronic Design*, pp. 262–267, March 2018.
 [19] Predictive Technology Model (PTM) website, <http://ptm.asu.edu/>.
 [20] LFRic, <https://www.metoffice.gov.uk/research/modelling-systems/lfric>.
 [21] Modelling and Prediction, <https://www.ecmwf.int/en/research/modelling-and-prediction>.