Energy-Efficient Time-Based Adaptive Encoding for Off-Chip Communication

DOI:
10.1109/TVLSI.2020.3018062

Document Version
Accepted author manuscript

Energy-Efficient Time-Based Adaptive Encoding for Off-Chip Communication

Eleni Maragkoudaki, *Student Member, IEEE* and Vasilis F. Pavlidis, *Senior Member, IEEE*

***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. A novel circuit is implemented which exploits the time domain to represent complex bit transition computations as delays and, thus, limits the power overhead due to encoding. The effectiveness of AWR is validated in terms of decrease in both bit transitions and power consumption. AWR is shown to yield higher power savings compared to three state-of-the-art techniques reaching 23% and 61% during the transfer of multiplexed address-data and image files, respectively, at just 1 mm wire length.**

***Index Terms*—Interconnects, Off-chip communication, Low-power, Energy-efficiency, Encoding scheme, Switching activity, Data transfer**

I. Introduction

THE ever growing compute density, enabled by parallelism, in combination with the non-decreasing system size has led to an excessive demand for data movement [1]. The energy cost of transferring data across the memory hierarchy is estimated between 18% and 40% of the total system energy for scientific applications and is expected to further increase in upcoming exascale systems [2]. In addition, the energy of transferring data does not scale as fast as the energy for computation [3]. For instance, the energy consumed for off-chip communication is estimated $115\times$ higher than that of an add operation in smart phone devices [4]. Thus, the power reduction of inter-chip communication is a key challenge for modern integrated systems.

An effective solution to decrease the dynamic power consumption is to reduce signal transitions through data coding. The benefits of using an encoding scheme (specifically the Bus Invert (BI) technique [5]) are investigated in [6]. When BI is applied to double-data-rate fourth generation (DDR4) memory, the I/O power as well as the power-supply noise are decreased.

Encoding schemes can achieve significant energy savings particularly when the switching behaviour of the data stream is known in advance. However, this may not be feasible or the characteristics of the data stream may vary over time. Therefore, this work extends our recent work [7], where an adaptive encoding scheme, namely Adaptive Word Reordering (AWR), is proposed for wide off-chip buses. AWR decreases the switching activity without affecting the communication bandwidth where the statistics of the data stream are not known *a priori*. The core idea of this technique is the reordering of the words of the data stream such that signal transitions are minimum. Optimal reordering requires a formidable amount of computations, therefore, a heuristic algorithm, namely Nearest Neighbour (NN) is preferred. The proposed scheme is envisioned for block data transfers, such as Direct Memory Access (DMA) where the data is transmitted only after a block is fully available and, thus, the latency introduced by encoding does not affect the performance of the communication. In this work, AWR is compared with state-of-the-art techniques in terms of total power savings including the power overhead of the encoder and decoder logic. Furthermore, the resilience of the proposed circuit implementation to process variations is explored and the related trade-offs between accuracy, power, and timing are discussed.

Although reordering data has been attempted in [8], the related algorithm does not fully exploit the potential of this approach in reducing switching activity as only the two most recent words are considered in the reordering operation. Consequently, the provided reduction in switching activity is comparable to BI which is shown to typically be inferior to AWR. Furthermore, [8] does not provide a circuit implementation, therefore, the circuit overhead in power is not considered. However, as shown in [9], the effectiveness of encoding in reducing power consumption is often restricted by the high implementation overhead, thereby diminishing and often eliminating the benefits offered theoretically by data encoding techniques. Hence, the efficiency of the encoding technique in [8] can not be evaluated properly.

On the contrary, in this work the NN search is considered for the reordering, which effectively decreases bit transitions. To the best of the authors' knowledge, NN has not been used for data encoding before. Additionally, a novel circuit that implements NN is proposed, which exploits the time domain for complex computations of Hamming distances, to decrease the power overhead of encoding. AWR decreases the power consumption of transferring data while the circuit power overhead is low, therefore, leading to significant power savings as compared to state-of-the-art techniques discussed in Section II.

The remainder of this paper is structured as follows. The proposed technique is described in Section III and is compared with competitive algorithms in terms of decrease in bit transitions. The circuit implementing this method is

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).

The authors are with the Advanced Processor Technology Group, School of Computer Science, University of Manchester, UK, (e-mail: eleni.maragkoudaki@manchester.ac.uk; pavlidis@cs.man.ac.uk).

described in Section IV. The power gains and limitations of the proposed technique are quantified in Section V and are also compared with existing encoding schemes. Finally, conclusions are drawn in Section VI.

II. Related Work

Encoding schemes for interconnect buses have been proposed aiming either latency [10] or power, where the majority of these methods target primarily to decrease power. In this paper, the focus is on the latter objective. Encoding schemes can be classified to either static or adaptive. Static schemes exploit the statistical properties of data streams, considered known at design time, and are discussed in subsection II-A. Alternatively, adaptive schemes assume that there is no prior knowledge of these properties, observe the data stream at runtime, and apply the appropriate encoding. These methods are described in subsection II-B.

A. Static Schemes

Techniques that exploit the sequential nature of signals in address buses belong to the static category. Gray code [11], for example, ensures a single bit transition when the data words are consecutive. T0 code [12] prevents transitions in case of sequential addresses using one extra bit line. Modified versions of this technique, T0-BI, Dual-T0, and Dual-T0-BI [13], are proposed to further reduce the switching activity of address buses. The Beach Solution [14] is an application oriented encoding scheme, based on prior analysis of the address stream.

In the case of data buses, the data words are not sequential, therefore, the previous techniques are rather not effective. A technique that is tailored to data buses and assumes that the statistical characteristics of the data stream are known at design time is the probability-based mapping [15]. The words with high probability of occurrence are mapped to code words with lower Hamming weight to decrease the overall transitions. This idea is implemented using reversible circuits in [16], where only part of the statistics is given. Algorithms that automatically generate low activity codes based on statistical information are presented in [17]. In [18], the Partial Bus Invert (PBI) method is proposed, where a subgroup of bus lines is composed according to the transition probabilities and the transition correlations among the bus lines. These lines are inverted to reduce transitions.

B. Adaptive Schemes

The efficiency of static techniques depends strongly on the behaviour of the application. Therefore, the benefits of static schemes diminish if the statistical properties of data vary temporally. To cope with this situation, several encoding techniques do not rely on *a priori* information of data statistics. Bus Invert (BI) is one of the early encoding schemes for decreasing transitions [5]. The data word is inverted if more than half of the bits switch. One extra bus line is used to inform the receiver whether the data word is inverted or not. This technique remains popular because of its effectiveness and simple circuit implementation. However, BI provides limited savings in switching activity compared to more elaborate techniques especially in cases where the data are highly correlated as shown in subsection III-B2.

The Adaptive Partial Bus Invert (APBI) is an extension of PBI, according to which the data stream is observed over time and the subgroup of encoded lines is changed periodically [19]. The effectiveness of this technique is restricted to highly correlated data streams such as image files. Bus Shifting (BS) rotates the word to be transmitted by a number of bits to minimise transitions and uses spatial redundancy to send this number to the decoder [20]. However, the calculation of Hamming distance for all possible rotations of each data word increases significantly the hardware complexity. The working zone encoding is well-suited to address buses and assumes that applications use specific subgroups of the address space [21]. Thus, this scheme is not suitable for data buses.

Frequent Value is an adaptive technique which encodes the frequently transmitted words and leaves the rest of the words unencoded [22]. However, a Content-Addressable Memory (CAM) is required in this case to store the frequent values, which increases the power and hardware requirements. Adaptive dictionary encoding [23] uses a dictionary to store recurring patterns to reduce the number of bits required to represent large patterns. The effectiveness of this technique is restricted to data with high correlation of adjacent bits.

Sequence-switch coding [8] is based on the reordering of words to reduce transition activity. The heuristic proposed in [8] considers only the two most recent words in the reordering operation and, therefore, the resulting decrease in switching is comparable to BI. This scheme, however, requires a more complex implementation than BI. Consequently, the overall performance of this technique is inferior to BI. Furthermore, the circuit architecture of this technique is not provided, therefore, the added circuit overhead can not be determined and the efficiency of this technique can not be fully and properly evaluated.

Data Exchange using Synchronized Counters (DESC) is a time-based technique where the data-values are represented by delays between consecutive clock pulses [24]. The limitation of this scheme is that the bandwidth is decreased by $3.15\times$ on average [25]. Adaptive Time-based Encoding (ATE) [25] is a technique that improves the energy-delay product by applying either DESC or binary encoding depending on the workload. Slow Transition Fast Level (STFL) signaling encodes data to mitigate the performance of low voltage swing wires in last level caches and, thus, balance bandwidth and power [26]. In [27], STFL is applied on off-chip interconnects and specifically different DRAM interfaces.

Energy Control and Metadata Consolidation are toggle-aware compression techniques that decrease the bit transitions caused by compression [29]. Hence, these techniques remove the negative effects on toggle rate of compression, however, do not decrease the switching activity of the initial data stream. Adaptive Bus Encoding (ABE) [28] is another adaptive scheme that observes the data stream thereby forming a cluster with the highly correlated lines of the bus. This scheme, although effectively decreases bit transitions, suffers from high overhead

in power as described in subsection V-B.

Several encoding techniques [30]–[33] focus only on the decrease in relative switching between adjacent bus lines, while the self-transitions of individual lines are not considered. Therefore, the applicability of these methods is restricted to those interconnects, where the coupling capacitance is dominant and the ground capacitance is not significant. Hence, these techniques are less effective for inter-chip interconnects considering the large capacitive load added from I/O cells.

Other recent works aim to minimise the energy consumed in modern memory interfaces, such as off-chip DRAM, which use asymmetric termination. The objective of these techniques is to reduce the number of 0's or 1's depending on the type of termination (VDDQ or VSSQ termination) rather than the switching activity or a combination of both. Data Bus Inversion can be adapted for these interfaces [34]. DC/AC Data Bus Inversion (DBI) [35] and DBI-AC/DC [36] are both extensions of BI. More is Less (MiL) uses sparse codes to reduce the number of 0's at the expense of longer data words [37]. Base+XOR Transfer [38] and Bitwise-Difference Encoding (BD-Encoding) [39] both exploit the similarity of data and Online Data Clustering groups data into different clusters based on their resemblance [40]. Alternatively, Cost-Aware Flip Optimization (CAFO) focuses on reducing the total cost of write operations in Phase Change Memory (PCM) and spin-transfer torque random access memory (STT-RAM) in terms of endurance and error rate, respectively [41].

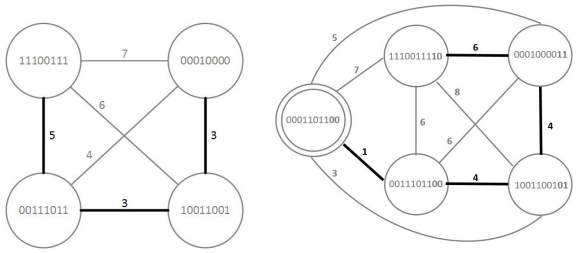
III. Adaptive Word Reordering

The proposed encoding scheme (AWR) is based on the observation of the data stream over a fixed window of N words and the dynamic reordering of these words in order to decrease the total number of transitions on the encoded bus. The proposed scheme can be applied to memory interfaces that do not use asymmetric termination such as LPDDR3 where power can be saved by reducing the number of bit transitions. Furthermore, access to LPDDR3 is burst oriented. Therefore, a block of read or write data is fully available at the beginning of the transmission which is required by the encoder of AWR in order to not introduce additional latency.

The process of identifying the order of the words that yields the minimum switching activity, as well as the Nearest Neighbour algorithm on which the proposed approach is based, are discussed in subsection III-A. In subsection III-B, an evaluation of the proposed technique is provided, where AWR is compared with state-of-the-art encoding schemes in terms of decrease in switching activity.

A. Algorithm

The problem of the optimal word reordering can be described by considering each word as a vertex of a complete, undirected, weighted graph $G(V, E)$. The weight of each edge, w, is the Hamming distance of each pair of words. The problem can be reformulated as the identification of the minimum weight route that visits all the vertices of the graph exactly once. This is very similar to the Travelling Salesman Problem (TSP) classified as NP-hard.



(a) Minimum weight route that visits all vertices exactly once.

(b) Route followed by the Nearest Neighbour algorithm. The search is assumed to start at the vertex 00011011.

Fig. 1: Examples of routes that reduce bit transitions.

The difference between the two problems is that the route in TSP is cyclic and has to end to the starting vertex. However, this difference does not change the computational complexity of the problem. An example of a graph is illustrated in Fig. 1(a), where the number of reordered words is $N = 4$ and the word length is $M = 8$ bits. The minimum weight route is highlighted.

The most straightforward solution to the TSP is an exhaustive search, for which the calculation of the cost of all possible rearrangements ($N!$) is required. Exact algorithms that determine the optimal solution and alleviate the computational cost have been developed [42]. However, the computational cost remains high and the hardware complexity of these algorithms is prohibitive if any savings in power are to be harvested.

There is a plethora of heuristic algorithms that determine near-optimal solutions to the TSP problem. In this work, the Nearest Neighbour search is implemented as a less complex approach. The NN algorithm is properly adapted for the specific problem. In each clock cycle, the search starts with the previously transmitted word and the weights with all the unvisited vertices (words that have not been transmitted) are calculated. To keep track of the order, a unique code of $K = \log_2 N$ bits is assigned to each word before transmission. In this way, data can be placed back in the initial order at the receiver side. These K bits are included in the calculation of the Hamming distance to ensure that these bits do not notably increase the switching activity. In the example of Fig. 1(b), the previous word is assumed to be 00011011, thus, the search starts from this vertex. The last two bits of each word are the order bits (shown with boldface). The route followed by the NN algorithm to reduce transitions is highlighted.

The pseudocode of the algorithm is shown in Fig. 2. N transmissions are required to transfer the N words over the interconnect. Therefore, index *i* is used to count the transmissions. While index *j* is used to iterate between all the words to find the one with the minimum Hamming distance for each transmission. If a word has been transmitted, then the Hamming distance is not calculated and the word is not considered for retransmission.

B. Performance Evaluation

The theoretical performance of the proposed reordering technique is determined by the switching activity, which is

TABLE I: Parameters of the investigated encoding techniques. The parameters are selected such that each technique yields the highest decrease in switching activity.

	AWR_{32}	AWR_{64}	BI	$APBI_1$	$APBI_{16}$	ABE_1	ABE_4
Observation window N	32	64	1	32	32	16	16
Bus width M	64	64	64	64	64	64	16
Update of observation window in clock cycles	32	64	1	32	512	17	17
Spatial redundancy in bits	5	6	1	1	1	1	4
Temporal redundancy in clock cycles	0	0	0	0	0	1	1

while Queue with words not empty **do**
  Update the N words to be reordered
  Assign a unique code of K bits to all N words
  for $i \leftarrow 0$ to $N-1$ **do**
    for $j \leftarrow 0$ to $N-1$ **do**
      if word[j] not transmitted **then**
        Calculate the Hamming distance between word[j] and previously transmitted word including the unique code of K bits
      end if
    end for
    Transmit the word with the minimum Hamming distance
  end for
end while

Fig. 2: Pseudocode of the NN algorithm amended to the requirements of AWR.

TABLE II: Characteristics of data streams.

Data Streams	Type	Size	Total Transitions Self	Relat.
LFRic	Address-data	2.44 MB	2,987,633	3,650,307
InfOli	Address-data	2.44 MB	2,961,115	3,537,631
LITE_LOOP	Address-data	2.44 MB	2,961,773	3,539,832
LU_SOLVER	Address-data	2.44 MB	2,962,321	3,540,071
NASTY_EXPS	Address-data	2.44 MB	2,961,212	3,539,298
Image	Data	790 KB	1,464,148	3,114,043
Random	Data	2.44 MB	5,120,736	10,079,981

compared with three other encoding schemes. These schemes are described in subsection III-B1 while the related results are presented in subsection III-B2.

1) Compared Techniques: Three state-of-the-art encoding techniques are selected that do not require *a priori* knowledge of data statistics and a circuit implementation is provided. BI [5] is a low power adaptive scheme that calculates the Hamming distance of consecutive data words and inverts the transmitted data word if the Hamming distance is higher than half of the word length. To indicate whether a word is inverted or not, one extra bus line is used. APBI [19] observes the data stream for a window of a fixed number of N words and forms a mask with the bus lines with the higher probability of switching. BI is then applied to these bus lines and one extra bit is used to inform the decoder about inversion.

ABE [28] selectively encodes a cluster of highly correlated bus lines. First, observes the data characteristics over a window of N words. Based on these characteristics a cluster is formed and the line with the maximum correlated transitions with the lines of the cluster is selected as the basis line. The lines in the cluster are finally XOR-ed with the basis. The basis and cluster information are transmitted using a redundant bus line and an additional clock cycle at the beginning of the window, respectively.

The parameters of the compared techniques APBI, ABE are specifically selected such that the techniques yield the highest reduction in switching activity according to [19] and [28]. In this way, a fair comparison is conducted that provides the highest savings for each technique. For all of the simulations, the bus width is considered to be $M = 64$ bits whilst the observation window is $N = 32$ words for APBI and $N = 16$ words for ABE. The mask computation of the APBI technique can be executed in each window of N words ($APBI_1$), but intervals of 16 windows ($APBI_{16}$) are also explored to further reduce the power consumption of encoding and decoding and provide a fairer comparison. Therefore, both of these scenarios are included. Furthermore, two cases for ABE are considered. In the first case, ABE is applied to the entire bus (ABE_1) while in the second case, the bus is split in 4 groups of $M/4$ bits and ABE is applied individually to each group (ABE_4) to best exploit this technique.

The decrease in switching activity of AWR is reported using both $N = 32$ (AWR_{32}) and $N = 64$ (AWR_{64}) reordered words. The savings increase with a higher number of reordered words; however, the power overhead of encoding and decoding increases faster. Therefore, 32 and 64 reordered words are selected as high savings in switching activity are provided while the power overhead is restrained. The parameters of all the encoding schemes as well as the related (spatial and temporal) redundancies are listed in Table I.

2) Savings in Switching Activity: The switching activities are determined for diverse data types, both synthetic and real, which are reported in Table II. The LFRic, InfOli, LITE LOOP, LU SOLVER, and NASTY EXPS benchmarks comprise a mix of memory addresses and data and are generated from different applications. LFRic [43] 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 [44]. Furthermore, an image file is used as well as uniformly distributed random data generated with Matlab.

The results of the decrease in switching compared to the

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

Data	AWR_{32}		AWR_{64}		BI [5]		$APBI_1$ [19][1]		$APBI_{16}$ [19][1]		ABE_1 [28][1]		ABE_4 [28][1]	
Streams	Self	Relat.	Self	Relat.	Self	Relat.	Self	Relat.	Self	Relat.	Self	Relat.	Self	Relat.
LFRic	36.88%	10.89%	**46.55%**	**25.20%**	14.37%	0.57%	13.48%	-13.18%	12.55%	-13.99%	17.91%	-0.78%	18.90%	-6.04%
InfOli	35.99%	10.55%	**47.99%**	**26.02%**	14.92%	0.54%	14.29%	-13.31%	13.45%	-14.73%	18.74%	-0.34%	19.80%	-5.68%
LITE_LOOP	36.02%	10.59%	**48.00%**	**26.06%**	14.91%	0.52%	14.30%	-13.35%	13.46%	-14.78%	18.74%	-0.32%	19.80%	-5.67%
LU_SOLVER	35.91%	10.61%	**48.00%**	**26.05%**	14.95%	0.53%	14.29%	-13.35%	13.46%	-14.73%	18.75%	-0.32%	19.81%	-5.66%
NASTY_EXPS	36.01%	10.58%	**47.99%**	**26.03%**	14.94%	0.52%	14.30%	-13.35%	13.47%	-14.74%	18.74%	-0.32%	19.80%	-5.67%
Image	72.99%	74.54%	**78.19%**	**79.78%**	22.58%	22.56%	35.07%	38.06%	28.17%	30.04%	38.93%	41.10%	32.51%	35.45%
Random	13.08%	13.03%	15.64%	15.53%	8.54%	8.50%	6.45%	6.48%	6.41%	6.46%	**16.66%**	**16.67%**	15.82%	15.32%

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

unencoded data streams are listed in Table III. In these results the spatial and temporal redundancies of each technique are also considered in the calculation of the switching activity. Moreover, in addition to the self transitions, the relative transitions are also listed as coupling is not negligible even for off-chip interconnects.

The AWR scheme provides greater self-switching savings for most types of data streams. For the multiplexed address and data streams, AWR_{64} yields a 46–48% reduction in bit transitions. The highest savings in switching activity are achieved for the image file with $\sim$78% reduction while the lowest are observed for the random data stream. In the latter case, ABE_1 provides the highest decrease in switching, yielding $\sim$17%.

The results of Table III highlight that the transitions contributed from the redundant bus lines of AWR add a low overhead as demonstrated by the high savings in bit switching for $N = 32$ and $N = 64$. For lower numbers of reordered words (N) the number of redundant bit lines decreases. However, this situation leads to lower savings in switching activity and, hence, in power. This trade-off is also demonstrated in the results of subsection V-C.

Relative transitions are calculated based on the transitions of adjacent bus lines according to Table 2 in [32] and can cause the charging or discharging of the coupling capacitance. The ground capacitance of off-chip wires is, typically, much higher compared to the coupling capacitance, therefore, the reduction of self transitions is crucial. However, the coupling capacitance is not negligible, hence, the impact of relative switching on power is notable. Although all of the techniques in Table III target minimising self transitions, relative transitions are also affected mostly in a positive way (i.e. transitions are reduced). The reduction in relative transitions of AWR_{64} range from $\sim$15% to $\sim$79%. The other techniques also provide high reduction for the image file and the random data. In case of the multiplexed address-data benchmarks, APBI and ABE cause a slight increase in the relative transitions. The likely cause of this behaviour is that whilst aiming to minimise self transitions, transitions of adjacent bus lines in opposite directions are produced. This type of transition dissipates more power compared to other switching events (e.g., relative transitions in the same direction) [31], [32]. Consequently, increasing the number of transitions in opposite directions can increase the the contribution of the coupling capacitance in the overall power figures.

IV. Proposed Circuit Architecture

The circuit architecture of the proposed encoding scheme is provided in this section. The encoder and decoder circuits are described in subsections IV-A and IV-B, respectively.

A. Encoder

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. The word with the lowest Hamming distance is then transmitted through the bus. This method requires N registers at the transmitter and the receiver to store the reordered words.

Conventionally, the computation of the Hamming distance is implemented using adder trees. This approach is highly inefficient in terms of power, especially for wide buses, counteracting any energy savings produced from encoding. Therefore, a different approach is followed, where the Hamming distance is determined as a delay in the time domain, drastically reducing the overhead in power due to encoding.

The proposed encoder circuit comprises three stages, as depicted in Fig. 3. 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 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.

The DEL signal that arrives first at the finish line of the race stage prevents the others from propagating. This condition is implemented in the finish stage. Before any signal arrives, a "0" is stored in all of 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. After all the DEL signals are reset, PER switches slowly back to 1 due to the weak pull-up resistor.

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 (i.e. these words yield the same number of switching bits), such that more than one of the SEL signals is equal to 1, the word with the lowest index is chosen.

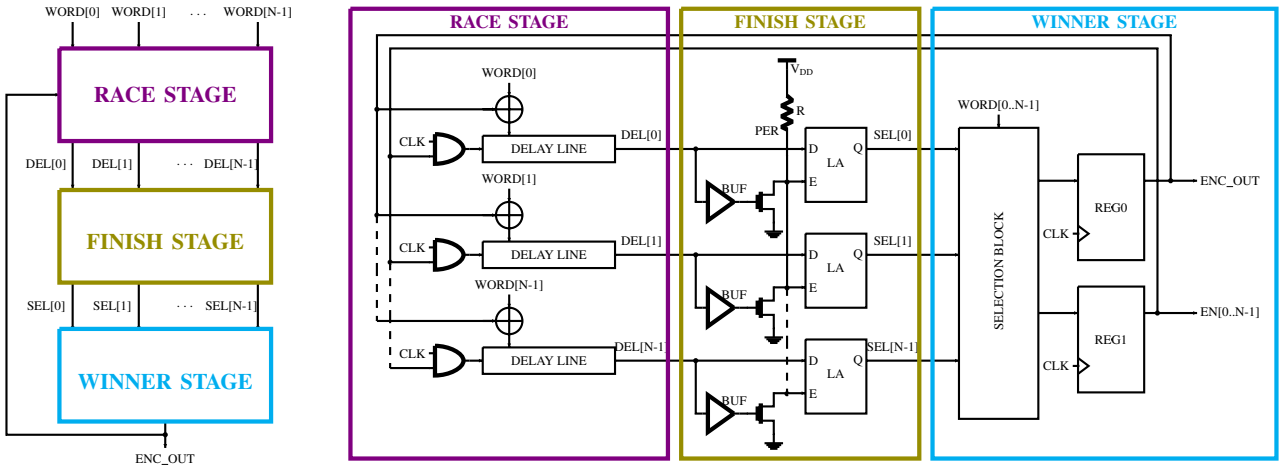


Fig. 3: Encoder circuit.

Transmitting the word with the lowest index can potentially decrease the delay of decoding. For example, if WORD[0] and WORD[7] arrive at the same time, there is no benefit in transmitting WORD[7] since the receiver can not utilise WORD[7] if all the previous words have not been read. The winning word is stored in the register REG0. To keep track of the transmitted words, a second register (REG1) is used, where the enable signals EN[0..N-1] are stored. Once the word to be transmitted is selected, the respective EN signal is switched to 0 and remains low until all N words are transmitted. All EN signals switch to high whenever a new block of N words is to be transmitted. EN signals enable or disable the respective delay lines allowing the propagation of the clock only for the words that have not been transmitted. To implement that an AND gate is used in the beginning of each the delay line. This mechanism not only prevents from transmitting a word twice but also saves dynamic power since the inverters in the delay lines of transmitted words do not switch.

An example of signal propagation with $N = 2$ is illustrated in Fig. 4. It is assumed that in the first clock cycle the Hamming distance of Word[1] is the lowest, therefore, DEL[1] is set to 1 faster than DEL[0]. DEL[1] causes both SEL[1] and PER signals to transition. Thus, SEL[1] is set to 1 and the latches are disabled before DEL[0] transitions to 1. PER switches slowly back to 1 after both DEL[1] and DEL[0] are reset. In the second clock cycle, the state of EN[1] changes to 0 as Word[1] was selected, while the clock pulse does not propagate through the delay line of Word[1], therefore, DEL[1] remains 0. Word[0] is selected in this cycle, since DEL[0] is the only signal that transitions to 1, causing SEL[0] and PER to flip. In the third clock cycle, both EN[0] and EN[1] are equal to 1 in order to enable the reordering of the next two words.

The delay line consists of a modified inverter chain as illustrated in Fig. 5, where W is the minimum width and the length is the minimum for all devices as determined by the utilised technology library. Each inverter is connected either to the ground or the supply voltage through a pair of devices. A detailed description of this delay line can be found in [45]. Briefly, when the i^{th} bit switches, T(i) switches to 1 and the

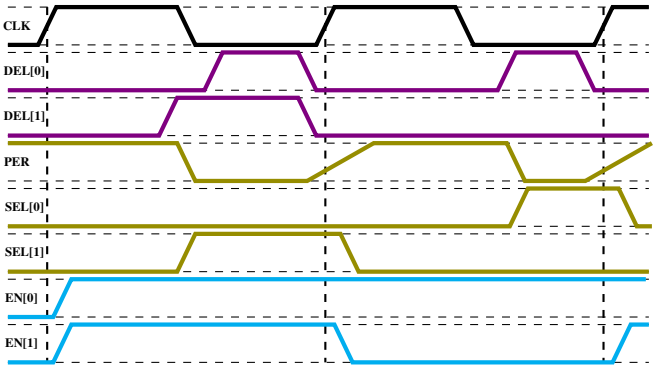


Fig. 4: Signal propagation of the encoder circuit.

i^{th} inverter is connected to ground (i is odd) or V_{DD} (i is even) through only one device. In case no transition takes place, the inverter is connected to either V_{DD} or ground through two devices connected in parallel. Hence, in the former case the delay is larger than in the latter case. The delay of only the first edge of each inverter (falling for even and rising for odd) is affected by the Hamming distance since the pair of devices is only added to either the pull-down (even inverters) or pull-up (odd inverters). The delay of the second edge of each inverter is constant. This implementation ensures that the delay of the falling edges of the DEL signals is constant. Therefore, the PER signal switches to 1 at a specific time in every clock cycle as this happens exactly when DEL signals switch to 0. Hence, the risk of a DEL signal switching to 1 while PER is still low is eliminated.

B. Decoder

The role of the decoder at the receiver side is to place the words back in the initial order. To achieve that, the decoder uses N registers to store the words in the right order as they arrive. The order of each word is transmitted by using low spatial redundancy. K more bits are added to each word that indicate the order. Thus, for N words, $K = \log_2 N$ additional bus lines are required. The decoder reads the K bits and enables the corresponding register to store the word

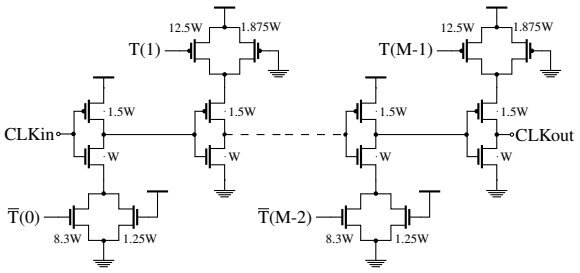


Fig. 5: Delay line circuit.

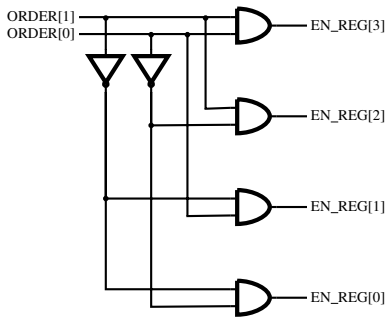


Fig. 6: Decoder circuit

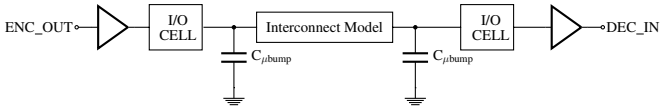


Fig. 7: Electrical model for an interposer-based interconnect.

TABLE IV: 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

while the rest of the registers remain disabled. The circuit that implements this function is a basic K-to-N decoder with each output connected to the enable input of the appropriate register. A 2-to-4 decoder circuit is depicted in Fig. 6 The K bits are also considered in the encoding process, thus, they are included in the calculation of the Hamming distance. Consequently, it is ensured that they do not considerably increase the power overhead.

V. Simulation Results

The power efficiency of the proposed AWR scheme is evaluated in this section. The simulation setup is described in subsection V-A. The overheads of AWR and the state-of-the-art encoding techniques are discussed in subsection V-B. Furthermore, the reduction in power of AWR is quantified in subsection V-C and is compared with the state-of-the-art techniques in subsection V-D. Finally, the robustness of AWR under process variations is investigated in subsection V-E.

A. Experimental Setup

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. However, note that the method is equally applicable to other state-of-the-art or emerging packaging approaches, such as Embedded Multidie Interconnect Bridge (EMIB) [46] and more broadly to applications that require wide and slow links. The link is assumed to connect two dies that are bump bonded on top of a silicon-based interposer. The circuit of the interconnect is illustrated in Fig. 7 and consists of the distributed wire model, I/O cells, and the parasitic capacitance of μbumps, $C_{\mu bump} = 30$ fF [47]. For the sake of simplicity, the mutual inductances are not considered. The global wire dimensions for a 65 nm technology are used according to [48]. The electrical characteristics of the wires for minimum pitch are listed in Table IV.

B. Circuit Implementation and Overheads

The encoder and decoder circuits of AWR as well as the other three techniques are implemented with a 65 nm technology [49]. The overheads in power and delay are measured using Spectre, Cadence® at nominal conditions (typical device corners, 27°C) and Design Compiler, Synopsys. The results of power consumption and delay are listed in Tables V and VI, respectively, where the bus is assumed to be 64 bits wide and the operating frequency is 400 MHz.

Note that to apply AWR to LPDDR3, the frequency has to be adjusted to 800 MHz. The frequency of AWR can be increased by implementing AWR in a more advanced technology as well as by decreasing the bus width. By narrowing the bus, the number of inverters in the delay line (Fig. 5) is considerably reduced since one inverter is required for each bit. The reduction of the critical path delay by reducing the bus width and, hence, the number of inverters is evident in the results of Table VI.

The power of the encoding and decoding circuits has a significant effect on the total power savings. A high overhead in power can diminish the efficiency of encoding and even increase power consumption [9]. BI exhibits the lowest overhead in power since the least amount of computations is required with this technique. Alternatively, ABE exhibits the highest power consumption, especially when applied to the entire bus (ABE_1) as the complexity of calculations increases exponentially with the number of bus lines. AWR and APBI have moderate power overheads.

To highlight the benefits of the time-based implementation of AWR, a digital implementation of the encoder of AWR with adder trees is described in Verilog and synthesized using Design Compiler, Synopsys. The power consumed by the encoder of this digital implementation for $N = 32$ and $M = 64$ is 20.13 mW on average. Thus, the conventional implementation consumes $\sim 5\times$ more power than the time-based encoder.

In terms of timing, AWR and ABE exhibit higher delays. However, the delay of AWR decreases significantly for lower bus widths since the delay lines are shorter. For example, for $M = 32$ bus lines and $N = 32$ the critical path delay is 1.6 ns. Furthermore, the latency of AWR ranges from 1 to N+1 cycles for the decoding of data. In the best case scenario where the words are transmitted in the initial order, the decoding latency

TABLE V: Power consumption of encoder and decoder circuits in mW.

Data Streams	Type	AWR_{32}		AWR_{64}		BI [5]		$APBI_1$ [19]		$APBI_{16}$ [19]		ABE_1 [28]		ABE_4 [28]	
		Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
LFRic	Address-data	3.76	0.01	6.25	0.02	0.79	0.39	3.26	2.61	1.45	0.88	49.27	2.29	14.30	1.95
InfOli	Address-data	3.29	0.01	5.62	0.02	0.67	0.35	3.01	2.43	1.34	0.81	48.17	2.29	14.04	1.94
LITE_LOOP	Address-data	3.29	0.01	5.62	0.02	0.67	0.35	3.01	2.44	1.34	0.81	48.17	2.29	14.04	1.94
LU_SOLVER	Address-data	3.29	0.01	5.62	0.02	0.67	0.35	3.01	2.44	1.34	0.81	48.17	2.29	14.04	1.94
NASTY_EXPS	Address-data	3.29	0.01	5.62	0.02	0.67	0.35	3.01	2.44	1.34	0.81	48.17	2.29	14.04	1.94
Image	Data	3.37	0.01	5.87	0.02	0.88	0.46	3.67	2.93	1.68	1.01	57.18	2.32	16.56	2.01
Random	Data	5.78	0.01	10.86	0.02	1.09	0.51	3.97	3.09	1.88	1.08	60.40	2.47	17.69	2.10

TABLE VI: Timing overheads of encoding techniques. The critical path delay is listed in ns and the latency in clock cycles. The bus width is $M = 64$ unless stated otherwise.

	AWR_{32}, M=32		AWR_{32}		AWR_{64}		BI [5]		$APBI_1$ [19]		$APBI_{16}$ [19]		ABE_1 [28]		ABE_4 [28]	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
Critical path delay	1.60	0.25	2.21	0.25	2.40	0.33	1.63	0.32	1.86	1.92	1.65	1.88	2.16	1.07	2.11	1.20
Latency	1 – N+1		1 – N+1		1 - N+1		1		1		1		N+1		N+1	
Bandwidth decrease	0		0		0		0		0		0		1/(N+1)		1/(N+1)	

is just 1 clock cycle, which is the time required to store the word in the respective buffer. In the worst case scenario, where the first word is transmitted last, the received words can only be utilised by the receiver after $N+1$ cycles after the first word is received and decoded, assuming that the order at the receiver matters. On the other hand, ABE always introduces a latency of N+1 cycles as the data are decoded after the information about the cluster and the basis line is received. The basis line information is only available after N cycles. The decoding requires 1 clock cycle, consequently, the total latency is N+1 cycles. BI and APBI increase latency by only one cycle. Finally, AWR, BI and APBI do not affect the communication bandwidth since 1 word is transmitted per clock cycle which is the same rate with the unencoded transmission. However, ABE requires N+1 cycles to transmit N words due to the temporal redundancy required to transmit the cluster information (see Table I). Therefore, ABE decreases the data rate by 1 word per N+1 cycles.

Often the energy-delay product is utilised to evaluate the efficiency of circuits. However, optimizing the energy delay product of the encoder of AWR is not straightforward. It is possible to decrease the delay but this decrease adversely affects the accuracy of the Hamming distance calculations as well as the power overhead. Techniques that reduce the delay in inverter chains such as tapering can not be applied in the delay lines since the delays of all the inverters should nominally be equal. Sizing up the devices uniformly in the delay lines can speed up the encoder, however, the accuracy of the delay lines as well as the power consumption are still affected. Particularly, increasing the width of the devices in the pull up and pull down that are always on (gate is connected to ground or VDD, respectively) results in a significant decrease of the total delay. However, this increase causes the difference in the delay of an inverter when a transition does and does not occur to shrink. This can lead to many DEL signals arriving at about the same time in the winner stage (see Fig. 3). As a result, the optimal word that exhibits the lowest Hamming distance is not always selected, thus, reducing the accuracy of the technique. On the contrary, sizing up the devices connected to $T(i)$ (see Fig. 5) mitigates this problem at the cost of higher power consumption. Finally, increasing the size of the inverters up to a certain value speeds up the whole inverter chain, nonetheless, power and accuracy are negatively impacted. By performing a parametric analysis with the widths of the devices, the specific combination of sizes illustrated in Fig. 5 is found to provide a balance between delay and power while accuracy is not compromised for the typical device corner.

C. Power Savings of AWR

The efficiency of AWR in saving power is evaluated for different interconnect lengths, bus widths (M), and number of reordered words (N). For this exploration, the LFRic benchmark is used, for which AWR_{32} yields about 37% and 11% decrease in self and relative switching, respectively. 1,600 words are transmitted over the interconnect, both encoded and unencoded and the overall power (i.e. encoder and decoder circuits and interconnect) is measured.

In Fig. 8, the percentage of power reduction is illustrated for a fixed bus width, $M = 64$ bits, and different number of reordered words and interconnect length. The power reduction is higher for larger N even though the circuit power overhead and the required spatial redundancy are higher. This result demonstrates that the power overhead of encoding and decoding is low for AWR compared to the high switching savings, yielding, in total, up to 23% power savings at just 1 mm wire length. Furthermore, the power savings increase for longer interconnects, which is expected since the capacitive load of wires increases, while the circuit overhead in power remains unaltered.

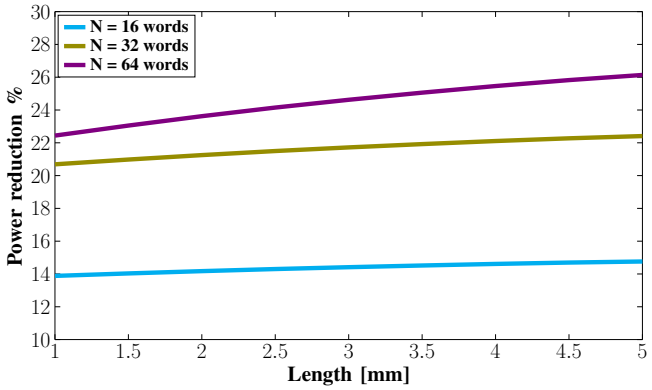


Fig. 8: Decrease in power for $M = 64$ bits and different number of reordered words N.

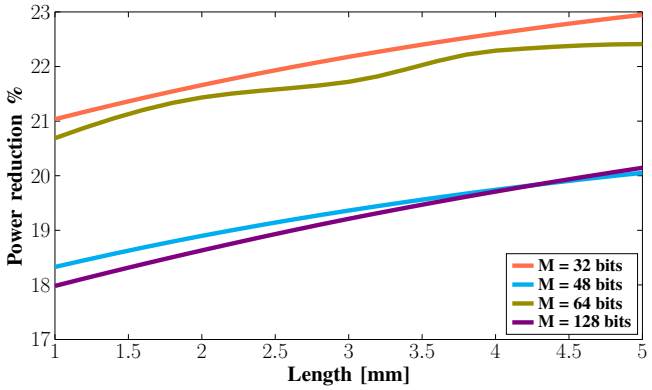


Fig. 9: Decrease in power for fixed number of reordered words $N = 32$ and different bus widths M.

AWR is also effective for different bus widths. This efficiency is demonstrated in Fig. 9, where the savings in power with respect to the bus width are illustrated. The power savings range from 18% to 21% at 1 mm and from 20% to 23% at 5 mm. Thus, the difference in savings is maximum 3% for different bus widths. Note that for $M = 128$ bits the frequency is reduced to 200 MHz due to the longer delay lines in the encoder circuit. However, the communication bandwidth is not affected considering that compared to the case of $M = 64$ bits, the bus width is doubled.

Note that for more advanced technologies, AWR yields higher power savings as encoding techniques are more beneficial due to low power consumed due to the smaller feature size of the transistors in these technologies. The power consumed in interconnects scales disproportionately compared to the power consumed in computations [3]. Therefore, the power overhead of encoding schemes decreases faster whilst the power consumed by interconnects decreases at a much slower pace as technology scales. Hence, the total power savings are expected to be even higher for advanced technologies.

D. Comparison with State-of-the-art

The proposed technique is compared with the discussed state-of-the-art schemes in terms of power savings in Fig. 10 for all the benchmark data of Table III. The bus width is $M = 64$ bits and the wire length is 1 mm. As expected the savings in power are lower than the theoretical savings of Table III for all the encoding schemes due to the added power overhead of the encoding and decoding circuits.

AWR outperforms the other techniques for all the real data streams as the high reduction in bit transitions greatly counteracts the low overhead in circuit power even at just 1 mm interconnect length. The low overhead is enabled by properly mapping the Hamming distances as delays in the time domain. The power savings of AWR for the multiplexed address-data benchmarks are ∼23% while for the image file, AWR improves power by ∼61%.

In contrast, the other techniques are less effective. BI yields low power savings compared to AWR ranging from 4.5% to 9% for the address-data benchmarks and reaches ∼22% for the image data. APBI is only effective for the image file where the data are highly correlated yielding up to 22% decrease in power. For the address-data benchmarks the savings of APBI_{16} range from -3% to 4%. The efficiency of BI and APBI is limited as the encoding is less effective in decreasing bit transitions compared to AWR and ABE. In other words, these methods emphasize mostly circuit simplicity, which is only one aspect of data encoding for power reduction.

Although ABE effectively reduces bit transitions, the high power overhead undermines the power savings. For the image benchmark, ABE_4 yields a mere 0.6% reduction in power while for all the other cases increases the power consumption by at least 18%. When ABE is applied to the entire bus (ABE_1), the power penalty is even greater due to the increased power of the encoder. Consequently, compared to BI and APBI, ABE places care mostly on the strength of the encoding, overlooking, however, the implications of this practice on the power of the encoding and decoding circuits. Alternatively, AWR carefully balances these two rather conflicting objectives, enabling higher savings in the total power from all of these techniques.

The benefits of data encoding diminish for random data. BI provides just 5% decrease in power while AWR_{32} and APBI_{16} only "break even". Thus, the power overhead of the encoder and the decoder is equal to the power saved in the interconnect. The rest of the techniques increase power. However, note that for longer interconnects the power reduction is higher as the power saved on the bus increases while the power overheads are the same.

From the results of Fig. 10, the conclusion that AWR provides higher savings for applications with more correlated data can be drawn. The efficiency of AWR is higher for image files where the data words have more similarities. In contrast, the bit transitions can not be significantly reduced by reordering the data words if the Hamming distance between all the most recent data words is high. Hence, AWR does not save power for random data.

In [39], the frequency of occurrence of 90% data similarity among the 64 most recent data words is measured for SPEC 2006 workloads [50]. Specifically, how often 58 bits or more out of a 64-bit word match exactly with the bits of one of the 64 most recent words is calculated. The frequency of 90% data similarity ranges between ∼22% and ∼100%.

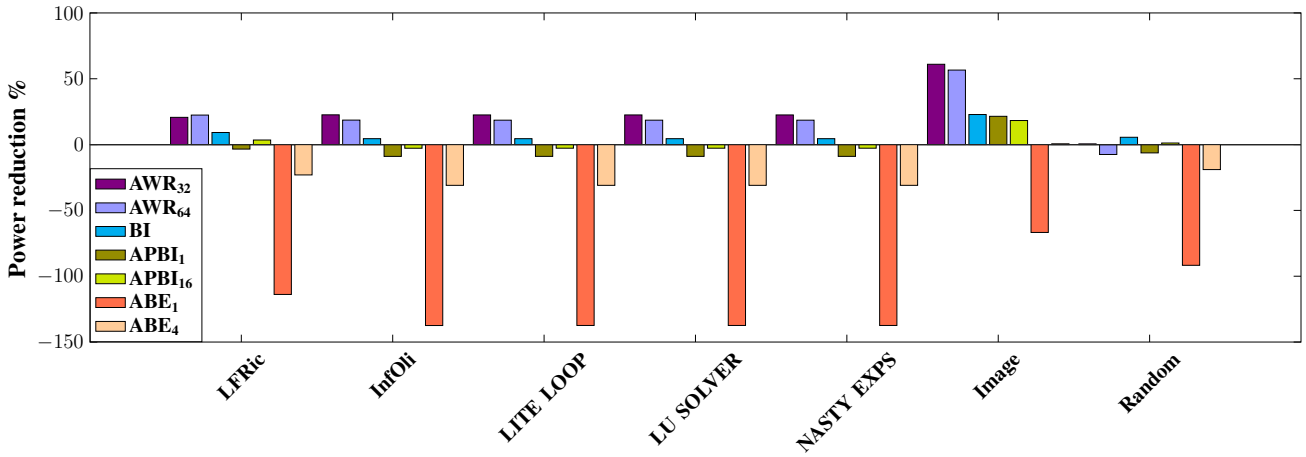


Fig. 10: Comparison of AWR with existing encoding techniques in terms of total power savings for a 1 mm long, 64-bit interconnect.

AWR would provide higher savings for applications with high frequency of occurrence of 90% similarity such as libquantum, a library for the simulation of quantum computers, omnetpp, a large ethernet simulator, soplex, a Simplex Linear Program (LP) solver, and zeusmp, a computational fluid dynamics code. Alternatively, the savings of AWR would be lower for bzip, a compression code, gromacs, a package for molecular dynamics, and milc, a simulator of four dimensional SU lattice gauge theory on MIMD parallel machine.

Finally, a detailed analysis of the power consumption is presented in VII using the LFRic and the Image benchmarks. The overheads of encoding and decoding as well as the power consumed in the interconnect are reported for all the techniques along with the power consumed in transmitting data unencoded. The lowest power consumption in the interconnect due to encoding is provided by AWR for both benchmarks and the circuit overhead of AWR is low, hence, the overall power gain is significant. BI and APBI are less effective in decreasing the interconnect power, however, the total power savings of BI are considerable because of the significantly low circuit overhead of this technique. Alternatively, ABE achieves a substantial decrease in the interconnect power for both benchmarks, however, the circuit overhead is high. Especially, in the case that ABE is applied to the entire bus (ABE_1), the overhead is higher than the power consumed in the interconnect for both address-data and data benchmarks, which increases the total power consumption. This behaviour is due to the fact that the circuit complexity of ABE increases quadratically with the number of bus lines.

E. Resilience of AWR to process variations

The robustness of the proposed encoder and decoder circuits under process variations and the effect of these variations on saving power are investigated in this subsection. The results of corner analysis are reported in Table VIII, where $N = 32$ words and $M = 64$ bits. The savings in power are reported both with and without including the overhead in power of the encoding and decoding circuits. AWR saves power for all the explored process corners, however, the efficiency of encoding

TABLE VII: Detailed analysis of power consumption using the LFRic and the Image benchmarks.

Encoding schemes	Overhead [mW]				Interconnect power [mW]			
	LFRic		Image		LFRic		Image	
	Enc.	Dec.	Enc.	Dec.	Enc.	Not enc.	Enc.	Not enc.
AWR_{32}	3.76	0.01	3.37	0.01	27.41	39.41	18.38	55.69
AWR_{64}	6.25	0.02	5.87	0.02	24.29	39.41	18.27	55.69
BI [5]	0.79	0.39	0.88	0.46	34.61	39.41	38.84	55.69
$APBI_1$ [19]	3.26	3.26	3.67	2.93	34.87	39.41	37.12	55.69
$APBI_{16}$ [19]	1.45	0.88	1.68	1.01	35.70	39.41	42.81	55.69
ABE_1 [28]	49.27	2.29	57.18	2.32	32.70	39.41	33.36	55.69
ABE_4 [28]	14.30	1.95	16.56	2.01	32.25	39.41	36.78	55.69

is lower in case of process variations since the savings on the bus are lower and the circuit power overhead is higher compared to the typical/typical (TT) corner.

The low drop of the savings on the interconnect (without including the circuit power overhead) is due to the delay lines (see Fig. 5). Particularly, for the SF and FS corners, the delay of the odd inverters (connected to ground through a pair of devices connected in parallel) deviates from the delay of the even inverters (connected to the supply voltage through a pair of devices connected in parallel). Therefore, the difference in delay resulting from the optimal word (i.e. the word with the lowest Hamming distance) and near-optimal words shrinks. Consequently, often a near-optimal word is selected which has a similar delay and, therefore, Hamming distance. This behaviour leads to a low increase in the switching activity of the transmitted data and, thus, a low drop in the power savings in the interconnect ($\sim$4%). However, the robustness of the link does not degrade as the encoder always selects a word to transmit even if two or more words have identical delays. Hence, one word is transmitted per clock cycle and, consequently, the throughput is not affected.

For the FF corner, the delay of the inverters in the delay line is shorter and, hence, the delay difference when a bit transition does and does not occur can not be sufficiently high

TABLE VIII: Process corner analysis results using the LFRic benchmark.

Corner	Overhead [mW]		Power savings	
	Encoder	Decoder	without circuit power	with circuit power
TT	3.76	0.01	30.26%	20.68%
FF	4.09	0.01	28.15%	17.70%
SS	5.54	0.01	26.35%	12.25%
SF	4.07	0.01	26.13%	15.77%
FS	4.11	0.01	26.13%	15.67%

to ensure that the "best" word (lowest Hamming distance) is always selected. Consequently, a mere $\sim$2% increase in switching activity compared to TT is observed. Finally, to meet the timing constraints for the SS corner, the devices in the delay line are sized up leading to shorter delays and higher power overhead. The shorter delays affect the accuracy of the calculation of the Hamming distances as in the case of the FF corner, leading to slightly higher switching activity.

Although AWR is less efficient in case of process variations, the savings in power ($>$12%) remain significant and the reliability of the communication is not affected. A way to mitigate the impact of process variations and maintain the savings on the interconnect is by regulating the delays of the delay lines such that the optimal word, i.e. the word with the lowest Hamming distance, is always the fastest. A way of regulating the delays is the sizing up of the devices of the delay lines. However, the deterioration of the efficiency of AWR in reducing the power consumed in the interconnect is just $\sim$4% in the FS and SF corners, where the difference in the delays of consecutive inverters is maximum. Consequently, the additional power overhead due to sizing up can surpass this low drop of 4%.

An alternative solution would be a self-calibration scheme to adjust the delay of each inverter and minimise the deviations. However, a self-calibration scheme would require additional circuits and routing resources. Considering that each stage of the delay line would require at least two devices to compensate for mismatch between the pull up and down network, the increase in the power of the delay line will be greater than the mere loss of 4% due to non-optimal word transmission. If the power of the additional circuitry required for the control of these calibrating devices is also incorporated in the power of the encoder, the savings in power will rather vanish. Hence, a more prudent approach is to design for the nominal case instead of the worst case, trading off a small reduction in the efficiency of the method with a significant increase in the circuit power overhead due to the sizing up of the devices or the addition of a self-calibrating circuit.

VI. Conclusion

In this work, an adaptive encoding scheme (AWR) is proposed for parallel off-chip interconnects. AWR decreases the interconnect power by changing the order of the transmitted words to effectively reduce switching activity. AWR is adaptive, thus, *a priori* knowledge of the statistical characteristics of data is not required. Notably, the circuit power overhead of AWR is restrained by proposing a novel encoder circuit, which exploits the time domain for the computation of Hamming distances by replacing power-hungry adder trees with delay lines.

AWR outperforms state-of-the-art techniques in terms of both decrease in switching activity and overall power savings for real data streams for several applications relating to high performance computing where energy reduction is a primary objective. AWR yields up to 23% and 61% savings in power for multiplexed address-data and image benchmarks, respectively, at just 1 mm interconnect length. Finally, the robustness of AWR is investigated under process variations. Although AWR is less efficient compared to the typical corner, the provided savings in power remain substantial where process variations are considered.

References

[1] S. Borkar, "Role of Interconnects in the Future of Computing," *Journal of Lightwave Technology*, Vol. 31, No. 24, pp. 3927–3933, December 2013.

[2] G. Kestor, R. Gioiosa, D. J. Kerbyson, and A. Hoisie, "Quantifying the Energy Cost of Data Movement in Scientific Applications," *Proceedings of the IEEE International Symposium on Workload Characterization*, pp. 56–65, September 2013.

[3] 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, November 2013.

[4] D. Pandiyan and C. J. Wu, "Quantifying the Energy Cost of Data Movement for Emerging Smart Phone Workloads on Mobile Platforms," *Proceedings of the IEEE International Symposium on Workload Characterization*, pp. 171–180, October 2014.

[5] M. R. Stan and W. P. Burleson, "Bus-Invert Coding for Low-Power I/O," *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, Vol. 3, No. 1, pp. 49–58, March 1995.

[6] H. Y To, "An Analysis of Data Bus Inversion: Examining Its Impact on Supply Voltage and Single-Ended Signals," *IEEE Solid-State Circuits Magazine*, Vol 11, No 2, pp. 31–41, June 2019.

[7] E. Maragkoudaki, P. Mroszczyk, and V. F. Pavlidis, "Adaptive Word Reordering for Low-Power Inter-Chip Communication," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 980–983, March 2019.

[8] M. Yoon, "Sequence-Switch Coding for Low-Power Data Transmission," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 12, pp. 1381–1385, 2004.

[9] C. Kretzschmar, A. K. Nieuwland, and D. Muller, "Why Transition Coding for Power Minimization of on-Chip Buses does not work," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 512–517, February 2004.

[10] P. P. Sotiriadis and A. Chandrakasan, "Reducing Bus Delay in Submicron Technology Using Coding," *Proceedings of the 2001 Asia and South Pacific Design Automation Conference*, pp. 109–114, 2001.

[11] 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, October 1994.

[12] L. Benini *et al.*, "Asymptotic Zero-Transition Activity Encoding for Address Busses in Low-Power Microprocessor-Based Systems," *Proceedings of Great Lakes Symposium on VLSI*, pp. 77–82, March 1997.

[13] L. Benini *et al.*, "Address Bus Encoding Techniques for System-Level Power Optimization," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 861–866, February 1998.

[14] L. Benini *et al.*, "System-Level Power Optimization of Special Purpose Applications: The Beach Solution," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 24–29, August 1997.

[15] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "A Coding Framework for Low-Power Address and Data Busses," *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, Vol. 7, No. 2, pp. 212–221, June 1999.

[16] R. Wille *et al.*, "Synthesis of Approximate Coders for On-chip Interconnects Using Reversible Logic," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1140–1143, March 2016.
[17] L. Benini *et al.*, "Architectures and Synthesis Algorithms for Power-Efficient Bus Interfaces," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, No. 9, pp. 969–980, September 2000.
[18] Y. Shin, S. Chae, and K. Choi, "Partial Bus-Invert Coding for Power Optimization of System Level Bus," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 127–129, August 1998.
[19] C. Kretzschmar, R. Siegmund, and D. Mueller, "Adaptive Bus Encoding Technique for Switching Activity Reduced Data Transfer over Wide System Buses," *Proceedings of the International Workshop on Integrated Circuit Design, Power and Timing Modeling, Optimization and Simulation*, pp. 66–75, September 2000.
[20] M. Alamgir, I. I. Basith, T. Supon, and R. Rashidzadeh, "Improved Bus-Shift Coding for Low-Power I/O," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 2940–2943, May 2015.
[21] E. Musoll, T. Lang, and J. Cortadella, "Working-Zone Encoding for Reducing the Energy in Microprocessor Address Buses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 6, No. 4, pp. 568–572, December 1998.
[22] J. Yang and R. Gupta, "FV Encoding for Low-Power Data I/O," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 84–87, August 2001.
[23] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, "An Adaptive Dictionary Encoding Scheme for SOC Data Buses," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1059–1064, March 2002.
[24] M. N. Bojnordi and E. Ipek, "DESC: Energy-Efficient Data Exchange using Synchronized Counters," *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pp. 234–246, December 2013.
[25] P. Behnam, N. Sedaghati, and M. N. Bojnordi, "Adaptive Time-based Encoding for Energy-Efficient Large Cache Architectures," *Proceedings of the ACM International Workshop on Energy Efficient Supercomputing*, November 2017.
[26] P. Behnam and M. N. Bojnordi, "STFL: Energy-Efficient Data Movement with Slow Transition Fast Level Signaling," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 1–6, June, 2019.
[27] P. Behnam and M. N. Bojnordi, "STFL-DDR: Improving the Energy-Efficiency of Memory Interface," *IEEE Transactions on Computers*, March 2020.
[28] 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 Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 25, No. 11, pp. 3057–3066, November 2017.
[29] G. Pekhimenko *et al.*, "A Case for Toggle-Aware Compression for GPU Systems," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, March 2016.
[30] K. Kim *et al.*, "Coupling-Driven Signal Encoding Scheme for Low-Power Interface Design," *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pp. 318–321, November 2000.
[31] M. Ghoneima and Y. Ismail, "Low Power Coupling-Based Encoding for On-Chip Buses," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 325–328, May 2004.
[32] Y. Zhang, J. Lach, K. Skadron, and M. R. Stan, "Odd/Even Bus Invert with Two-Phase Transfer for Buses with Coupling," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 80–83, August 2002.
[33] P. Subrahmanya, R. Manimegalai, V. Kamakoti, and M. Mutyam, "A Bus Encoding Technique for Power and Cross-talk Minimization," *Proceedings of the IEEE International Conference on VLSI Design*, pp. 443–448, January 2004.
[34] K. Sohn *et al.*, "A 1.2 V 30 nm 3.2 Gb/s/pin 4 Gb DDR4 SDRAM With Dual-Error Detection and PVT-Tolerant Data-Fetch Scheme," *IEEE Journal of Solid-State Circuits*, vol. 48, No. 1, pp. 168–177, January 2013.
[35] J. Lucas, S. Lal, and B. Juurlink, "Optimal DC/AC Data Bus Inversion Coding," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1063–1068, March 2018.
[36] T. M. Hollis, "Data Bus Inversion in High-Speed Memory Applications," *IEEE Transactions on Circuits and Systems–II Express Briefs*, Vol. 56, No. 4, pp. 300–304, April 2009.
[37] Y. Song and E. Ipek, "More is Less: Improving the Energy Efficiency of Data Movement via Opportunistic Use of Sparse Codes," *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pp. 242–254, 2015.
[38] D. Lee, M. O'Connor, and N. Chatterjee, "Reducing Data Transfer Energy by Exploiting Similarity within a Data Transaction," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, February 2018.
[39] H. Seol *et al.*, "Energy Efficient Data Encoding in DRAM channels exploiting Data Value Similarity," *Proceedings of the IEEE/ACM International Symposium on Computer Architecture*, June 2016.
[40] S. Wang and E. Ipek, "Reducing Data Movement Energy via Online Data Clustering and Encoding," *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, October 2016.
[41] R. Maddah, S. M. Seyedzadeh, and R. Melhem, "CAFO: Cost Aware Flip Optimization for Asymmetric Memories", *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 320–330, February 2015.
[42] 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, June 1992.
[43] LFRic, https://www.metoffice.gov.uk/research/modelling-systems/lfric.
[44] Modelling and Prediction, https://www.ecmwf.int/en/research/modelling-and-prediction.
[45] M. Fujino and V. G. Moshnyaga, "An Efficient Hamming Distance Comparator for Low-Power Applications," *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, pp. 641–644, September 2002.
[46] R. Mahajan *et al.*, "Embedded Multidie Interconnect Bridge - A Localized, High-Density Multichip Packaging Interconnect," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, No. 10, pp. 1952–1962, October 2019.
[47] P. Mroszczyk and V. F. Pavlidis, "Ultra-Low Swing CMOS Transceiver for 2.5-D Integrated Systems," *Proceedings of the IEEE International Symposium on Quality Electronic Design*, pp. 262–267, March 2018.
[48] Predictive Technology Model (PTM) website, http://ptm.asu.edu/.
[49] United Microelectronics Corporation (UMC), http://umc.com/.
[50] Standard Performance Evaluation Corporation, SPEC CPU 2006, http://www.spec.org/cpu2006.