Adaptive Word Reordering for Low-Power Inter-Chip Communication

Document Version
Accepted author manuscript

Link to publication record in Manchester Research Explorer

Citation for published version (APA):

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester’s Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.
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 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.

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).
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.

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 \( \mu \) 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>
<thead>
<tr>
<th>( R ) [( \Omega/mm )]</th>
<th>( L ) [( \mu H/mm )]</th>
<th>( C_{GND} ) [( \mu F/mm )]</th>
<th>( C_{G} ) [( \mu F/mm )]</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.74</td>
<td>1.52</td>
<td>222.55</td>
<td>52.45</td>
</tr>
</tbody>
</table>

![Fig. 1: Encoder circuit.](attachment:fig1.png)
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_EXP 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.

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Self Relative</td>
<td>Self Relative</td>
<td>Self Relative</td>
<td>Self Relative</td>
</tr>
<tr>
<td>LFRic</td>
<td>Address-Data</td>
<td>36.88% 10.89%</td>
<td>14.37% 0.57%</td>
<td>16.27% -12.83%</td>
<td>18.50% -5.73%</td>
</tr>
<tr>
<td>InFoli</td>
<td>Address-Data</td>
<td>35.99% 10.55%</td>
<td>14.92% 0.54%</td>
<td>15.72% -13.09%</td>
<td>18.12% -5.74%</td>
</tr>
<tr>
<td>LITE_LOOP</td>
<td>Address-Data</td>
<td>36.02% 10.59%</td>
<td>14.91% 0.52%</td>
<td>15.75% -13%</td>
<td>18.13% -5.66%</td>
</tr>
<tr>
<td>LU_SOLVER</td>
<td>Address-Data</td>
<td>35.91% 10.61%</td>
<td>14.95% 0.53%</td>
<td>15.65% -13.01%</td>
<td>18.08% -5.62%</td>
</tr>
<tr>
<td>NASTY_EXP</td>
<td>Address-Data</td>
<td>36.01% 10.58%</td>
<td>14.94% 0.52%</td>
<td>15.75% -13%</td>
<td>18.12% -5.69%</td>
</tr>
<tr>
<td>BW image</td>
<td>Data</td>
<td>37.28% 39.04%</td>
<td>10.03% 0.18%</td>
<td>15.93% 34.31%</td>
<td>32.56% 41.67%</td>
</tr>
<tr>
<td>Colour image</td>
<td>Data</td>
<td>72.99% 74.54%</td>
<td>22.58% 22.56%</td>
<td>37.88% 40.96%</td>
<td>36.02% 36.02%</td>
</tr>
<tr>
<td>Random</td>
<td>Data</td>
<td>13.08% 13.05%</td>
<td>8.54% 8.50%</td>
<td>10.59% 11.02%</td>
<td>6.17% 7.53%</td>
</tr>
</tbody>
</table>

$^1$Minus sign implies an increase in the number of transitions.
$V_{DDCORE}$ of the used technology, the decrease in power is remarkable and reaches 23% for address-data buses for just 1 Mbit and different bus widths $N$.

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

<table>
<thead>
<tr>
<th>Data Streams</th>
<th>Interconnect Power</th>
<th>Overhead Power</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unencoded AWR BI</td>
<td>AWR BI</td>
<td>AWR BI</td>
</tr>
<tr>
<td>LFric</td>
<td>39.41</td>
<td>27.49</td>
<td>1.75</td>
</tr>
<tr>
<td>InFoil</td>
<td>32.40</td>
<td>21.78</td>
<td>1.29</td>
</tr>
<tr>
<td>LITE_LOOP</td>
<td>32.41</td>
<td>21.81</td>
<td>1.29</td>
</tr>
<tr>
<td>LU_SOLVER</td>
<td>32.41</td>
<td>21.81</td>
<td>1.29</td>
</tr>
<tr>
<td>NASTY_EXPS</td>
<td>32.41</td>
<td>21.81</td>
<td>1.29</td>
</tr>
<tr>
<td>BW image</td>
<td>12.02</td>
<td>8.137</td>
<td>0.75</td>
</tr>
<tr>
<td>Colour image</td>
<td>55.70</td>
<td>18.33</td>
<td>1.78</td>
</tr>
<tr>
<td>Random</td>
<td>58.26</td>
<td>52.16</td>
<td>5.54</td>
</tr>
</tbody>
</table>

**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).

**References**


