Computer Life (CPL) ISSN: 1819-4818
DELIVERING QUALITY SCIENCE TO THE WORLD

# An Energy Saving Method for On-chip Embedded Data Bus

Mingquan Zhang

School of Control and Computer Engineering, North China Electric Power University, Baoding

071003, China

---

*Abstract: In the paper, the author proposed a method named MC(Multi-Coding) that reduces the bus dynamic energy efficiently. In the proposed scheme, four encodings are induced. Experimental results show that the proposed encoding reduces the bus dynamic energy by an average of 10.5%, compared to which without the method, and by an average of 4.7%, compared to conventional bus invert coding.*
*Keywords: Bus energy, multi-coding, energy saving.*

---

## 1. INTRODUCTION

Switching activity (SA) on data transmission gives the number of times the bus lines are discharged and recharged between 0 and 1, and is directly responsible for the dynamic energy consumption on the data bus. The low power encoding techniques [1] are widely used to reduce SA for dynamic energy consumption and the effects of crosstalk (signal noise, delay) during data transmission on buses. They aim to transform the data being transmitted on buses in such a manner so that the self-SA and coupling SA on buses are reduced.

Bus invert encoding [2] proposed by M. Stan et al. is the most well-known to perform very well in the arena of energy reduction over the data bus. In bus invert encoding, a value is sent as it is, or in a bit-inverted form, depending on the state of the bus during the previous transaction. The main idea is to reduce the dynamic energy consumption by reducing the SA of the bus. Based on this, the researchers presented the odd/even invert encodings, the partial invert encoding and other extended forms [3, 4]. However, an encoding scheme is used alone will cause the inactive or unrelated bits unnecessary switching, which reduces the effects of data bus energy saving. Due to the data characteristic is different in different stages of applications and changing with different applications, the single encoding scheme is insufficient to meet the goal of optimizing the data bus energy saving. These encoding schemes have some flaws as follows: (a) They are not considered the influence of coupling SA, which accounts for a large proportion of total SA and leads to large number of bus energy consumption in deep sub-micron (DSM) technology. (b) The ratio of energy incomings to the introducing cost is not high with one encoding scheme alone. (c) The scope of the encodings is limited. To solve these problems, in this paper we design a method named MC, which can reduce the bus SA and bus dynamic energy efficiently, and introduce low overhead. The MC method introduces four encoding schemes: swap encoding, invert encoding, odd invert encoding and even invert encoding, which has some advantages compared with other bus energy saving encodings as follows: (a) Its

structure is simple and the energy consumption of itself is low; (b) The coupling SA and energy consumption introduced by it are considered; (c) It can automatically select the encoding scheme according to the SA number of different applications and different stages of the same application.

## 2. THE PROPOSED ENCODING METHOD

Our encoding structure includes two parts: encoding and decoding (Figure 1 and Figure 2 show their structures, respectively). The proposed method, using algebraic and permutation encoding schemes, introduces four encoding schemes: Swap encoding, Invert encoding, Odd invert encoding and Even invert encoding, which can effectively reduce the data bus SA total distance of a data transmission, and so the bus dynamic energy consumption is effectively reduced. When the value enters the encoder, it calculates the total SA distance and automatically selects the encoding scheme, which can obtain the minimum SA distance, and generates the encoded value and coding number to be sent. In order to ensure the receiver can correctly decode the transmitted data, our method needs two extra control indication lines, which indicate the scheme to be used.

### 2.1 Encoding structure

We denote $B^j$ is the value to be sent on the $n$-bit width data bus at cycle $j$, and it can be expressed as $B^j = \left(b_n^j, b_{n-1}^j, b_{n-2}^j, \cdots, b_1^j\right)$, $B^{(j-1)enc}$ represents the encoded value at cycle $j$-1. Four encoding schemes of the scheme are expressed as: Swap $B^{(swap)}$, Invert $B^{(inv)}$ , Odd Invert $B^{(odd)}$, Even Invert $B^{(even)}$ with the coding number 00, 11, 01, 10, respectively. Swap the adjacent bits of $B^j$ and append its coding number, we get the encoded value $B^{j(\text{swap})}$.

The encoded value that all the bits of $B^j$ are inverted then appended its coding number is defined as $B^{j(inv)}$. The encoded value that the odd bits of $B^j$ are inverted then appended its coding number is defined as $B^{j(odd)}$. The encoded value that the even bits of $B^j$ are inverted then appended its coding number is defined as $B^{j(even)}$. The coding number indicates which encoding scheme is used to encode the sending value. $X$ is used to calculate the vertical distance between the two $n$-bit width values; $Y$ is used to calculate the horizontal distance of n-bit width value.

Figure 1 shows the encoder schematic diagram. The input value with $n$-bit width is entered the four encoding components at the same time and they produce four types ($n$+2)-bit encoded values. In addition, subsequently, the encoded values are sent to distance estimator, in which obtained their total SA distance, respectively. In order to reduce the delay, each encoding process and distance evaluation are treated with at the same time. Finally, the distance comparator is responsible for selecting the encoded value $B^{j(enc)}$ which has the minimum SA total distance and producing coding number codenum.

### 2.2 Decoding structure

Figure 2 shows the decoder schematic diagram. The decoder is mainly consisted of one 2-4 decoder and four decoding circuits. The input of the 2-4 decoder is the control signal coding numbers a and b. The output of the 2-4 decoder produces one of the four decoding signals controlling one following decoding component to be valid. The valid component decodes to obtain the $n$-bit original value $B^j$. The four decoding components are: Swapper for swapping adjacent bits of the encoded value $B^{j(swap)}$ , Inverter for inverting all the bits of the encoded value $B^{j(inv)}$, Odd inverter for inverting all the odd bits

of the encoded value $B^{j(odd)}$ and Even inverter for inverting all the even bits of the encoded value $B^{j(even)}$ .

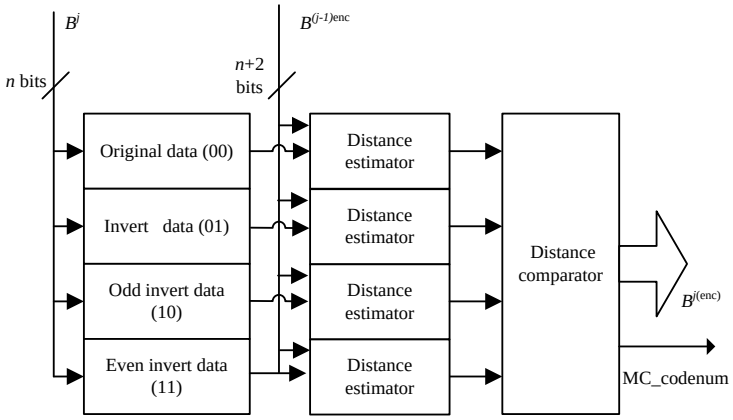


Figure1 MC encoder schematic diagram

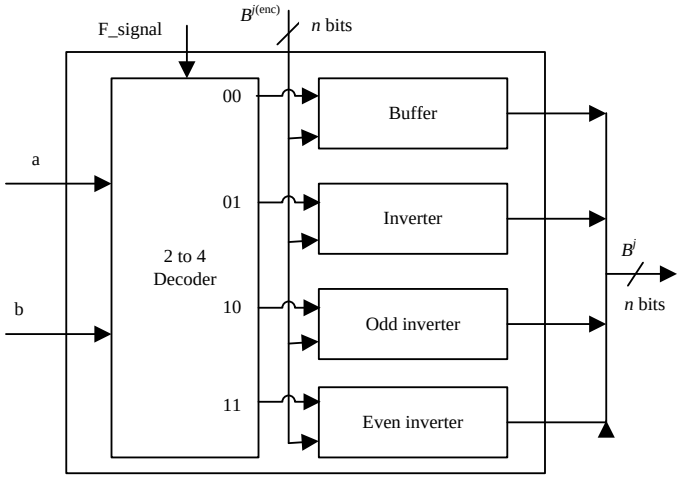


Figure 2 MC decoder schematic diagram

## 3. EXPERIMENT AND RESULTS ANALYSIS

### 3.1 Simulation environment

Our multi-core structure based on bus is four-core CMP structure, each core has 4-way set associative private first level of instruction cache (IL1) and data cache (DL1) with a size of 32KB, and 16-way shared second level cache (L2) with a size of 1MB. The cache row size at all levels is 64B, and it is included cache. The specific parameters are shown in Table 1. Each of the private L1 split caches is write-through. The shared L2 cache is write-back and maintains inclusion with respect to the L1 cache. In our multi-core system, the cores and L2 cache are connected by bus. The data bus is alternately used by different cores, so as to achieve the purpose of access the shared L2 cache.

To quantify the effect of dynamic bus energy saving with our method, we have carried out simulation experiments under the CMP structure with our encoding on the Archimulator, which is a multi-core architectural simulator of our research group. The default simulation parameters that have been used in the simulation experiments are given in Table 2. We have used three memory-intensive benchmarks with helper threads(ht) from the Olden [5] and CPU2006 [6] suites: mst ht, em3d ht, and 429.mcf ht, representing typical tasks that might be present in a multi-core system. All applications

were executed with the default input sets provided with the benchmarks suites. All three benchmarks are cross-compiled using GCC at O3 optimization level.

Table 1. Three Scheme comparing

| parameters | value |
|---|---|
| cores | 4 |
| IL1/DL1 size | 32KB |
| L1 associativity | 4-way |
| bus width | 32+2 lines |
| bus energy/access | 11.6pJ/line |
| Coupling factor λ | 5 |

### 3.2 Analysis of dynamic energy saving effect

Through the simulation experiments, we compare the effect of reducing bus energy consumption under different measures. The measures we have conducted are: exclusively using swap encoding(SWAP); exclusively using invert encoding (INV); exclusively using odd invert encoding(ODD); exclusively using even invert encoding(EVEN) and our method (MC). We use (2) to measure the energy saving effect of each measure. With 70nm technology, we compare the effect of the above various measures on the dynamic energy saving when coupling factor λ is equal to 5.

$$\mathrm{E} = (\mathrm{X} + \lambda \cdot \mathrm{Y}) \cdot C_L \cdot V_{DD}^2 \quad (1)$$

$$\delta = \left(1 - \frac{E_{enc}}{E_{org}}\right) \times 100\% \quad (2)$$

The energy saving detailed results are shown in Table 2. From Table 2 we can see that the effect of energy saving is not obvious when each of other measures is used, whose maximum ratio of energy saving is 5.86%, this is because the incomings from using a single measure offsets its own energy consumption. However, when the MC method is used, we can get the maximum energy saving ratio is 11.3% (em3d ht), and the average energy saving ratio can be reached 10.53 %. This is because the MC encoder can perceive the SA number and automatically select the encoding scheme for minimizing the total SA distance, which makes the SA be reduced more greatly. According to (1), the percentage of the reduced SA distance directly determines the dynamic energy saving effect of the data bus. And the MC method, which has obtained the maximum reduction of bus total SA distance, makes the energy saving effect the best.

In this paper we also carry out some other experiments, which are introduced more encoding schemes than MC, the results show that their effect is not significantly improved than MC on bus dynamic energy saving. However, they need to add more hardware units and control lines than MC which will increase the on-chip area cost and their own energy consumption. So the Table 2 Dynamic energy saving ratio comparison under different measures (λ=5) MC composed of four encoding schemes is a better option for on-chip embedded data bus energy saving.

Table 2. The ratio of dynamic energy saving with different measures (%)

| | δ_SWAP | δ_ODD | δ_EVEN | δ_INV | δ_MC |
|---|---|---|---|---|---|
| mst_ht | 3.24 | 5.21 | 5.11 | 4.40 | 10.12 |
| em3d_ht | 4.02 | 5.72 | 5.86 | 4.96 | 11.30 |

| | | | | | |
|---|---|---|---|---|---|
| 429.mcf_ht | 3.35 | 5.31 | 5.34 | 4.87 | 10.18 |
| average | 3.54 | 5.41 | 5.44 | 4.74 | 10.53 |

## 4. CONCLUSION

In this paper, we investigate the dynamic energy saving of on-chip embedded data bus with the influence of coupling capacitance in DSM technology. We design an on-chip multi-core structure with bus energy saving modules, and present a new method called MC to optimize the dynamic energy saving of data bus. The MC method can perceive the SA number and automatically select the encoding scheme for minimizing the total SA distance. The results of simulation experiments show that our method can significantly reduce the total SA distance and effectively improve the bus dynamic energy saving ratio by about 10.53% in 70 nm CMOS technology. Compared with other methods, the MC method can be always obtained a better effect on data bus dynamic energy saving.

## ACKNOWLEDGEMENTS

This paper was financially supported by “the Fundamental Research Funds for the Central Universities (2018MS073)”.

## REFERENCES

[1]. Kaushik, Brajesh Kumar, Agarwal, Deepika,Babu, Nagendra G. Bus encoder design for reduced crosstalk, power and area in coupled VLSI interconnects[J]. Microelectronics Journal, 44(9):827-833 (2013).

[2]. Stan, M. R., Burleson, W. P., Bus-invert coding for low-power I/O[J], IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 3(1), 49-58 (1995).

[3]. Chiu, Ching-Te, Huang, Wen-Chih, Lin, Chih-Hsing, et al. Embedded transition inversion coding with low switching activity for serial links [J]. IEEE Transactions on Very Large Scale Integration (VlSI) Systems, 21(10):1797-1810 (2013).

[4]. Zhang M, Gan Z, Zhang J, et al. “Joint Hybrid Frequent Value Cache and Multi-Coding for Data Bus Energy Saving”, IEEE, International Conference on Parallel and Distributed Systems. IEEE, 869-876(2017).

[5]. Rogers, A., Carlisle, M. C., Reppy, J. H., Hendren, L. J., Supporting dynamic datastructures on distributed-memory machines[J], ACM Transactions on Programming Languagesand Systems, 17(2), 233-263 (1995).

[6]. Henning, John L., SPEC CPU2006 benchmark descriptions [J], SIGARCH Computer Architecture News, 34(4), 1-17 (2006).