

A Low Energy Method Based on FEV for Embedded on-Chip Data Bus

Mingquan Zhang

School of Control and Computer Engineering, North China Electric Power University, Baoding
071003, China

Abstract

The on-chip buses consume considerable amount of total energy of embedded multi-core chip in the deep submicron technology domain. In the paper, the author proposed a method based on FEV named FEVCBI (Frequent Exchange Value Cache-Bus Invert) encoding that reduces the data bus dynamic energy further than the conventional FV or bus invert coding. In the proposed scheme, only a small cache and one line are added to the bus. Experimental results show that the FEVCBI encoding reduces the bus dynamic energy by an average of 26%, compared to which without the method.

Keywords

Bus energy, frequent exchange value, bus invert coding.

1. Introduction

Energy consumption is one of the major aspects in the design of on-chip multi-core circuits. With the continuous scaling of silicon technology, area and power consumption of interconnects are one of the main bottlenecks for on-chip bus. As the technology scales down to deep submicron (DSM) technology, the on-chip bus dissipates a significant fraction of the total system power budget [1]. For this reason, the design of power efficient data bus is today recognized as a key issue, especially for the embedded multi-core on-chip system.

Dynamic energy consumption of on-chip bus is the main source of embedded multi-core chips. It is produced by capacitance charge and discharge caused by zero and one conversion in data transmission. Reducing energy consumption of on-chip bus currently has two main directions: one is to reduce the amount of communication on the bus; the other is to reduce the bit switching activity of transmission on the bus.

In recent years, the researchers have proposed some technologies to reduce the energy consumption of on-chip interconnection, including serial communication, data coding, changing the topology structure, increasing the auxiliary cache and so on [2-3], which have made some effects on reducing the energy consumption of on-chip multi-core interconnection. However, there are some problems as follows:

- (1) Need to add more extra hardware, the complexity of hardware is high.
- (2) Need to add on-line monitoring, the complexity of time is high.
- (3) Most of the technologies are for high performance multi-core computers, and the special needs of embedded multi-core chips is less considered.

In order to solve the above problems, we mainly study the bus dynamic energy consumption optimization of the embedded multi-core chips. Using the locality of exchange value, the optimization method based on FEVC is proposed, and the dynamic energy consumption of the on-chip bus is further reduced by combining with the bus invert coding. Our methods are irredundant, meaning that they require a small cache and one line to be added to the bus.

2. FEVCBI Encoding Scheme

2.1 FEVC Structure design.

The FEVC of this paper adopts the structure of communication value buffer in [2]. As shown in Figure 1, fvEN is used as the indicator signal to indicate whether the original value is transmitted on the link or its index in FEVC (the location number of the stored value). FEV_i is a 32-bit register for storing the i-th record in FEVC. FEVC uses content address structure to store four FEVs. In order to facilitate retrieval, encoding and decoding, the data values stored in FEVC are the same as the values in each FEVC, and remain unchanged throughout the operation of the program. The reason for this choice is: the coverage of the first four FEVs is higher; FEVC with large content address structure will increase search time and index number, which will increase energy consumption. Fixed FEVC content can reduce the data consistency band.

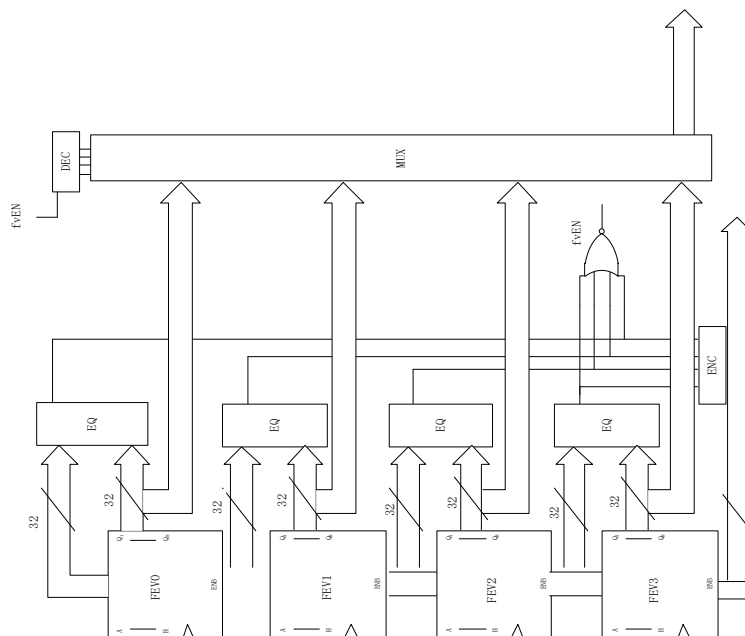


Figure 1 FEVC structure diagram

The principle of FEVC encoding is as follows: the data with low frequency is transmitted on the bus in the form of original value, and the high frequency data is transmitted on the bus in the form of encoded value, which has little SA. When the transmission data occurs between the L1-L1, the sender first quickly searches for whether the data value to be sent is in the FEVC of the L1. If the value is found in the FEVC, the index of the stored value in the FEVC is sent instead of the original value, and set the indication line indicating that the sending value is the index of the original value. The indication line is a control bit line that is added to the FEVC. If the sender end L1 does not find the value to be sent in FEVC, no replacement is made and the original value is sent. At the receiver end, when the indication line signal is observed, L1 uses the transmitted index of the value to find the original value in the FEVC. Without the indicated signal, the receiver L1 handles the original value directly. Such a data transmission between the L1-L1 is completed. As for this it can reduce the number of bit switching activity and so the bus energy consumption is reduced. When transmission data occurs between L1-L2, the same strategy is used to make use of FEVC.

We pipeline the whole mechanism, i.e. the lookup for the second value of the block in the FEVC is performed concurrently while the first value transfer goes on the bus. Similarly, at the receiver end, the lookups are pipelined. Consequently, the performance ramifications of a block transfer are not significantly affected. This mechanism may add two cycles (one at the sender and receiver each) overall per block for data communication, and with larger block sizes this overhead can become even less significant. It is easy to expand a CMP with n cores, it would have n + 1 FEVCs overall—one at each of the cores, and the working principle is the same.

2.2 Bus Invert coding.

Our approach is closely related to the bus invert (BI) coding [4]. Therefore, we first study the effect of BI coding on SA reduction. SA reduction rate with BI coding is as follows: For a sequence of w -bit code words, assume that their Hamming distances are h_1, h_2, \dots, h_n ,

$$h_i = \sum_{j=1}^w S_{(i-1)j} \oplus S_{ij}, \quad i = 1, 2, 3, \dots, n \tag{1}$$

where n is the length of the code sequence, s_{ij} the j th bit of word i (denoted by s_i) in the sequence, and \oplus the logic XOR operation.

Without any bus encoding, the total number of bit SA for the sequence of data when it is transferred on the bus is

$$SA = \sum_{i=1}^n h_i \tag{2}$$

When BI is applied to this sequence, some words will be bit-inverted, if their Hamming distances are larger than $w/2$, the half of word width. The associated Hamming distances will be changed accordingly. When BI encoding is taken into account, the Hamming distance of a word, s_i , can be generalized as

$$H_i = \begin{cases} h_i & c_{i-1} = 0, \\ w - h_i & c_{i-1} = 1, \end{cases} \tag{3}$$

where c_{i-1} is the invert control of the previous transfer; when it equals 1, the previous transferred value is bit inverted.

2.3 The Proposed FEVCBI Encoding

Combined with the characteristics of the above two encodings, we propose a method named FEVCBI to increase the auxiliary cache, that is, increasing FEV caches for frequent exchange values. With the features of FV coding and value locality, we add FEVC at sender and receiver ends. When the sent value is searched in the FEVC of the sender end, it is transmitted as the index in the FEVC, and the receiver end is prompted with an indicator line. When the sent value is not in the sender FEVC, the original value will be sent. At the same time, we combine with BI coding, the SA is further reduced, and the receiver end decides to deal with the received values according to the indicator lines.

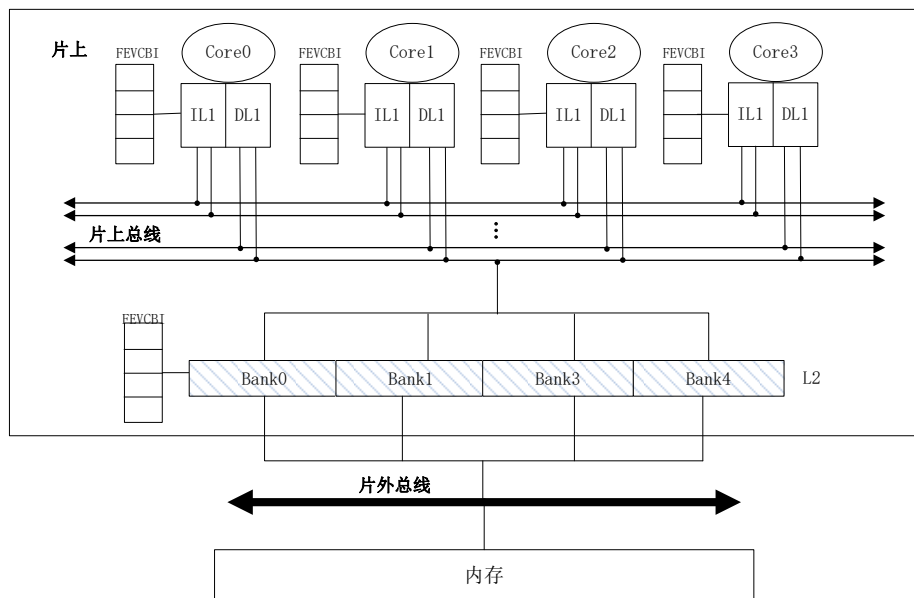


Figure 2. Multi-core architecture with FEVCBI based on bus

Our multi-core structure based on bus is as shown in Figure 2. It 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. In the structure, there are five FV cache modules - one at each of the cores and one at the L2 end as shown in Figure 2. 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.

3. Experiment and Results Analysis

The dynamic energy consumption of the on-chip bus is mainly derived from the charge and discharge of the capacitance on the lines, which is brought by the one and zero conversion. Of course, there are other factors that directly or indirectly affect the energy consumption. In this paper, we optimize the energy consumption of on-chip bus only by optimizing the transmission of data values. In order to simplify the modeling complexity, we temporarily ignore the energy consumption caused by the interaction capacitance between adjacent lines. The formula 1 and formula 2 are used to calculate the energy consumption of on-chip bus. Formula 3 measures the energy saving effect.

$$E_C = aV_{DD}^2 \times C_{wire} \times \sum_{i=1}^N \sum_{j=1}^M S_{i,j} \quad (1)$$

$$E = E_C + E_F \quad (2)$$

$$\varphi = \left(1 - \frac{E_X}{E_O}\right) \times 100\% \quad (3)$$

Where E_C represents bus lines energy consumption, E_F represents energy consumption of FEVC and the extra indicated lines, E represents the total energy consumption of on-chip bus, a is a factor, V_{DD} represents the power supply voltage, C_{wire} is bit line capacitance, and $S_{i,j}$ represents the number of switching activity on the bit line i from cycle j to cycle $j + 1$. And N represents the number of lines to transmit data. M represents the number of cycles of the program running. φ represents the energy saving ratio. E_O and E_X indicate the energy consumption of the on-chip bus before and after adopted measures respectively. When no FEVCBI is used, E_F equals 0. Experimental default parameters are shown in Table 1. To verify the efficiency of bus energy saving, some of the parameters are adjusted in the experiments. According to the paper [2], the energy consumption of each access of FEVC is 18.6pJ, and the energy consumption per line of the bus is the bit line energy consumption when the half of the bit line need to invert. The energy consumption of FEVC is also taken into consideration when evaluating the energy saving effect.

Table 1. Default Simulation Parameters

parameters	value
cores	4
IL1/DL1 size	32KB
L1 associativity	4-way
FEVC	4entries,fully-associative
FEVC energy/access	18.6pJ
bus width	32+2 lines
bus energy/access	11.6pJ/line

We select four programs from Mibench [5] for performance evaluation benchmarks: dijkstra, stringsearch, rijndael, qsort. The programs are cross-compiled with GCC for MIPSII executable files. In order to reduce the impact of code optimization on application performance during program design, and make the program reach its peak as far as possible when the program runs, the GCC optimization option is the best optimized -O3.

Table2. The ratio of bus energy saving with different measures

(%)	dijkstra	stringsearch	rijndael	qsort	average
$\varphi(E_c)$	16.64	13.96	5.84	16.53	13.24
$\varphi(E_f)$	-1.55	-1.5	-1.5	-1.51	-1.51
$\varphi(BI)$	15.09	12.46	4.34	15.02	11.73
$\varphi(E_c)$	25.62	26.76	15.1	27.92	23.85
$\varphi(E_f)$	-5.58	-6.05	-5.47	-6.28	-5.85
$\varphi(FEVC4)$	20.04	20.71	9.63	21.64	18
$\varphi(E_c)$	28.08	29.73	16.68	30.95	26.36
$\varphi(E_f)$	-8.86	-9.8	-8.64	-10.26	-9.39
$\varphi(FEVC8)$	19.22	19.93	8.04	20.69	16.97
$\varphi(E_c)$	38.74	38.15	17.94	41.71	34.14
$\varphi(E_f)$	-6.9	-7.58	-6.63	-7.67	-7.2
$\varphi(FEVCBI)$	31.84	30.57	11.31	34.04	26.94

As is known to all, bit SA directly determines the on-chip bus energy consumption. After using our method, the number of bit switching activity of the benchmarks is reduced in varying degrees, which brings advantages to bus energy saving. Table2 is shown the energy saving effect on bus with different measures. $\varphi(E_c)$ presents positive earnings of some measure and $\varphi(E_f)$ presents cost of some measure. $\varphi(BI)$ presents the bus energy saving ratio of using BI alone. $\varphi(FEVC4)$ and $\varphi(FEVC8)$ present the bus energy saving ratio of using FEVC with 4 FEVs and 8 FEVs, respectively. When FEVCBI with 4 FEVs is used, the average bus energy saving ratio is about 26%. The results indicate the proposed method is efficiency for data bus energy saving.

4. Summary

In this paper we proposed a bus energy saving method FEVCBI, which efficiently reduces the number of bus SA and so the bus dynamic energy is reduced. Experimental results show that the proposed encoding method is efficient.

Acknowledgements

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

References

- [1]. Jafarzadeh, N., Palesi, M., Khademzadeh, A., Afzali-Kusha, A., Data encoding techniques for reducing energy consumption in network-onchip[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2014, 22(3):675-685.
- [2]. LIU, C., Sivasubramaniam, A., Kandemir, M. Optimizing bus energy consumption of on-chip multiprocessors using frequent values[J]. Journal of Systems Architecture, 52(2):129-142(2006).
- [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]. Stan, M. R., Burleson, W. P., Bus-invert coding for low-power I/O[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1995, 3(1):49-58.

- [5]. M. R. Guthaus, J. S. Ringenberg, D. Ernst, et al. MiBench: A free, commercially representative embedded benchmark suite [C]. IEEE International Workshop on Workload Characterization, IEEE, 2001:3-14.