Bus Dynamic Energy Saving of Embedded Multi-core Chips Based on Frequent Value

Mingquan Zhang

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

mqzhang@ncepu.edu.cn

Abstract

To reduce the bus dynamic energy of embedded multi-core chips, we proposed a method of energy saving based on Frequent Value (FV). A Frequent Value Cache (FVC) is produced using the value locality of the bus, and the traffic and the bit switching activity on the bus is reduced, so the energy consumption of on-chip bus is efficiently reduced. To achieve the best effect of energy saving, the number of stored values in FVC was tested by the experiment. Compared with the original system, the results show that our solution can realize energy saving 13% when the FVC is exclusively used with four data values, and the energy saving proportion can be reached 18.7% when it is combined with the invert coding algorithm.

Keywords

Frequent Value Cache; bus; energy saving.

1. Introduction

Simple and efficient is the advantage of rapid development of embedded multi-core chips. However, the characteristics of battery power make energy consumption a bottleneck restricting its development. Therefore, when designing multi-core chips, energy consumption becomes a major consideration. Optimizing the energy consumption of embedded multi-core chips has become a hot spot in the research of embedded multi-core systems. Wang et al. [1] and PAUL et al. [2] respectively study the optimization of caching energy consumption for embedded multi-core chips. With the increasing number of cores on the chip, the energy consumption of on-chip bus is increasing, and the proportion of the chip consumption is becoming more and larger. These factors make it imperative to optimize the bus energy consumption of embedded multi-core chips.

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 [3-5], 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 data value, the optimization method based on FVC is proposed, and the dynamic energy consumption of the on-chip bus is further reduced by combining with the invert coding.

The main contributions of this paper are as follows: (1) the on-chip multi-core structure based on bus is designed. (2) The FVC is added to the multi-core structure, and the bus energy saving method of embedded multi-core chips is realized with the invert coding. The experimental results show that the proposed method is feasible and effective for the bus energy saving of embedded multi-core chips.

2. Energy Saving Method Design of Bus on Embedded Multi-core Chips

2.1 Multi-core Structure Platform Based on Bus

This multi-core structure based on bus is as shown in Figure 1. 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 FVC modules - one at each of the cores and one at the L2 end as shown in Fig. 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. When different cores need to access the same L2 bank at the same time, there will be competition, and a corresponding arbitration mechanism is needed to ensure the integrity and consistency of the data.

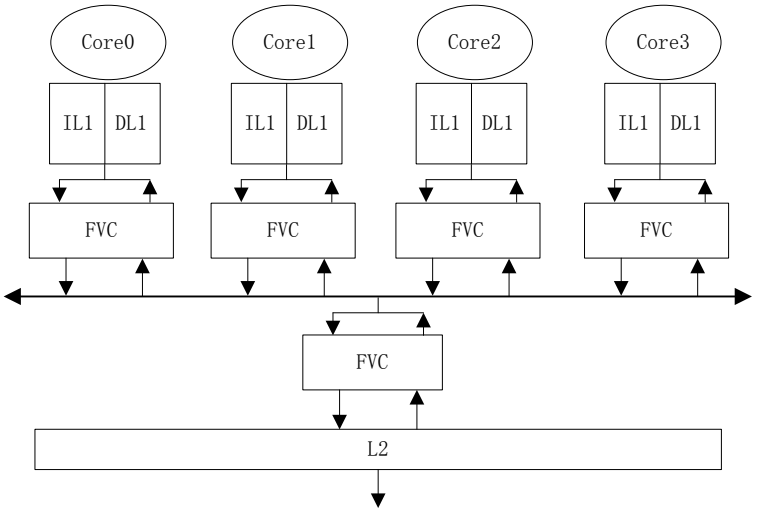

Fig.1 Multi-core architecture with FVC based on bus

2.2 The Frequent Value on the Bus

In the multi-core platform, the data dependency of the program context environment and the limited cache size, especially, the limited size of the first level cache leads to the memory miss, which determines the necessity of the multi-core data transmission on the chip. When executing, programs will show different degrees of temporal and spatial locality, so that some data values appear frequently in communication, which is frequent values (FVs). Under the multi-core structure shown in Fig. 1, three storage intensive programs of Olden[6] and CPU2006[7] benchmarks: em3d_ht, 429.mcf_ht, and mst_ht are used to evaluate the performance.we get the FVs by taking many times experiments on

the structure. Fig. 2 shows the coverage of the first n values that occur frequency, which is the percentage of FVs occupied all communicating data values in the program. As the figure shows, the coverage of the FVs is high; the average coverage of the first 4 values is up to 18%. In the subsequent discussion, we will discuss the characteristics of FVs based on Fig. 2.

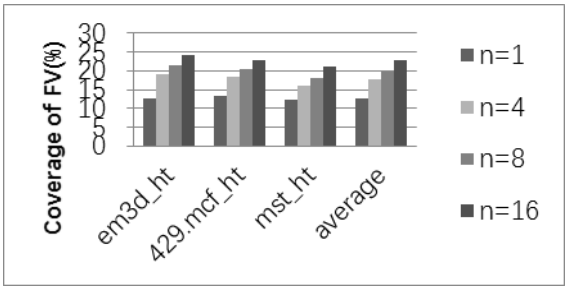

Fig.2 The coverage of FVs in data transmission

2.3 FVC Working Principle

In order to achieve bus energy saving optimization, we design FVC. The structure, as shown in Fig. 1, adds one FVC for each L1 cache and shared L2 cache, and increases 5 FVCs in our four-core structures with one each core and one for the L2 end (assuming two levels cache on the chip), and the values stored in each FVC are obtained at pre run.

In our structure, we use directory based on MSI protocol to ensure data consistency of cache. 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 FVC of the L1. If the value is found in the FVC, the index of the stored value in the FVC 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 FVC. If the sender end L1 does not find the value to be sent in FVC, 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 FVC. 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 FVC.

We pipeline the whole mechanism, i.e. the lookup for the second value of the block in the FVC 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 FVCs overall—one at each of the cores, and the working principle is the same.

2.4 FVC Combined with Invert Coding

Invert coding algorithm: the number of needing bit invert of the data value D^{k+1} is recorded as N^{k+1}, the number of needing bit invert of the data value $D^{k+1(INV)}$ is recorded as $N^{k+1(INV)}$. Inverting each bit of the (kth+1) time transmission value D^{k+1} we obtain the value $D^{k+1\,(INV)}$, then compared with the number of N^{k+1} and $N^{k+1\,(INV)}$, if $N^{k+1} \leq N^{k+1\,(INV)}$ the D^{k+1} is sent, otherwise $D^{k+1\,(INV)}$ is sent and set the indication line, indicating the receiver end whether the received data is the original value or its invert value. The receiver determines how to deal with the data value according to the indication line signal. When the indication line has been set, the data value received is processed by the invert code, otherwise, it is processed as the original value. In order to maximize the energy saving of the on- chip bus, we combines the FVC and the invert coding algorithm.

3. Bus Energy Consumption Model

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.

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

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

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

In which E_C represents bus lines energy consumption, E_F represents FVC energy consumption, 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. *Eo* and *Ex* indicate the energy consumption of the on-chip bus before and after adopted measures respectively. When no FVC is used, E_F equals 0. Experimental default parameters are shown in Table 1. According to the paper [6], the energy consumption of each access of FVC 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 FVC is also taken into consideration when evaluating the energy saving effect.

Table.1 Default experimental parameters

parameters	value
process technology	70nm
clock speed	500MHz
cores	4
IL1/DL1 size	32KB
L1 associativity	4-way
FVC	4entries,fully-associative
FVC energy/access	18.6pJ
bus width	32+2 lines
bus energy/access	11.6pJ/line

4. Experiment and Results Analysis

4.1 Benchmarks

The default parameters are shown in Table 1. To verify the efficiency of bus energy saving, some of the parameters are adjusted in the experiments. We select three storage intensive programs from Olden[6] and CPU2006[7] for performance evaluation benchmarks: mst_ht, em3d_ht and 429.mcf_ht. 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.

4.2 Analysis of Energy Saving and Optimization Effect

The amount of communication and bit switching activity on the bus has a significant impact on the bus energy consumption. In particular, bit switching activity directly determines the on-chip bus energy consumption. After using our method, the amount of communication and the number of bit switching activity are reduced in varying degrees, which brings advantages to bus energy saving. In order to achieve maximum energy saving, we compared with the bus energy saving effect for different n of FVC. According to the above formula, the energy consumption ratio under different measures is calculated as shown in Table 2. First lines indicate different measures, namely, F1, F4, and F8 respectively indicate the effect of n taking 1, 4, 8 only with FVC; f1inv, f4inv, f8inv respectively indicate the effect of n taking 1, 4, 8 by FVC and invert coding.

Table 2. The results of energy saving with different measures

benchmarks		ratio of energy consumption/%					
		f1	f4	f8	f1inv	f4inv	f8inv
mst_ht	E_F/E	2.16	4.37	5.80	2.16	4.37	5.80
	E_C/E	89.92	85.72	84.34	83.46	79.56	78.19
	φ	7.92	9.91	9.85	14.38	16.07	16.01
em3d_ht	E_F/E	2.11	4.90	6.47	2.11	4.90	6.47
	E_C/E	88.56	82.12	80.08	81.60	76.39	73.37
	φ	9.33	12.98	13.45	16.29	18.71	20.15
429.mcf _ht	E_F/E	2.30	5.00	6.45	2.30	5.00	6.45
	E_C/E	88.95	83.19	81.43	81.88	77.32	74.68
	φ	8.75	11.81	12.12	15.81	17.68	18.87

From Table 2, it is known that when FVC is used alone, the greater the n is, the larger the FVC energy consumption and the smaller the bus energy consumption. This is because the larger the n is, the more FV is stored in the FVC, and the energy of FVC is increased, while the number of data communication is reduced on the lines and the number of bit switching activity is reduced, and so the energy consumption of the bus is reduced.

As a whole, when the measure is only FVC, the bus energy consumption is reduced with the increase of n, and the maximum ratio of energy saving can be achieved 13.45% (em3d_ht, n=8). When FVC is used with the invert coding, the change trend of FVC and bus energy consumption is consistent with the single use of FVC, and the entire interconnection energy consumption is further reduced. However, the overall interconnection energy consumption varies with n. The trend of change is not monotonically decreasing, for example, the energy saving ratio of mst_ht with measure f4inv is 16.07% and the energy saving ratio with f8inv is 16.01%, this is because the increase of n makes FVC more energy consumption, which is more than the energy consumption income that it brings to the bus.

From the above analysis, the greater the value of n, the greater the energy consumption of FVC, and the limited space on the chip, so the value of n should not be too large. From Table 2, when the energy saving ratio of each benchmark of n=4 and n=8 is almost equal, mst_ht with n=4 (16.07%) even has a better energy saving ratio than n =8 (16.01%). In addition, the larger the n, the longer the time spent for each search. Therefore, considering the FVC taking with n=4 and combining with the invert coding (it is f4inv) as a measure to optimize the energy consumption of on-chip bus, the energy saving effect can be maximized. Fig. 3 is a comparison diagram of the energy saving effect on bus with different measures.

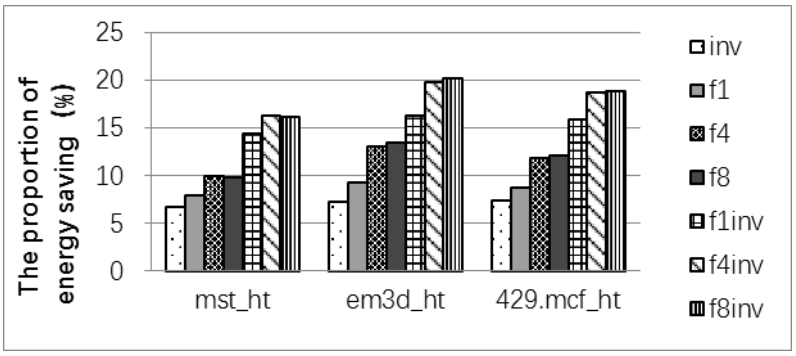

Fig.3 The proportion of energy saving with different measures

5. Conclusion

In this paper, using the locality of data value on the bus lines, the FVC is designed by obtaining the FV in the data communication of the CMP system. It is used to reduce the number of switching activity on the bus lines, and the on-chip bus energy saving optimization based on the FVC is realized. In order to investigate the sensitivity of FVC size to energy saving efficiency, we analysis the energy consumption with FVC in different sizes. The increase of FVC size (n) can further reduce energy consumption. However, considering the limited space of the chip and the performance cost of increasing the size, we select the size n=4 of FVC finally. The experimental results show that: compared with the original, the bus energy consumption can reduce 13% when only FVC is used. When combined with the invert coding algorithm, the energy saving ratio can reach 18.7%.

Acknowledgements

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

References

[1] WANG, W. X., Mishra, P., Ranka, S. Dynamic cache reconfiguration and partitioning for energy optimization in real-time multi-core systems[C]// Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference (DAC), San Diego, USA, 2011: 948-953.

[2] PAUL, M., Petrov. Adaptive I-cache partitioning for energy-efficient embedded multitasking [J]. IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, 2011, 19(11):2067-2080.

[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, 2013, 21(10):1797-1810.

[4] Asaduzzaman, Abu, Chaturvedula, Sri R., Pendse, Ravi. A novel folded-torus based network architecture for power-aware multicore systems [J]. Computers and Electrical Engineering, 2013:39(1)2494-2506.

[5] LIU, C., Sivasubramaniam, A., Kandemir, M. Optimizing bus energy consumption of on-chip multiprocessors using frequent values[J]. Journal of Systems Architecture, 2006, 52(2):129-142.

[6] Rogers, A., Carlisle, M. C., Reppy, J. H., et al. Supporting dynamic data-structures on distributed-memory machines [J]. ACM Transactions on Programming Languages and Systems, 1995, 17(2):233-263.

[7] Henning, John L. SPEC CPU2006 benchmark descriptions [J]. SIGARCH Comput. Archit. News, 2006, 34(4):1-17.