TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 8057 - 8062

Improving the Performance of Chip Multi Processor using Nano Caching Scheme

R. Arun Prasath , Asso. Professor, Department of ECE, Siddhartha Institute of Technology & Sciences, Narapally, Hyderabad.
R. Velmani, Asso. Professor, Department of ECE, Siddhartha Institute of Technology & Sciences, Narapally, Hyderabad.
E. John Alex, Asso. Professor, Department of ECE, CMR Institute of Technology, Medchal Hyderabad.
G. Sudhagar, Professor, Department of ECE, Siddhartha Institute of Technology & Sciences, Narapally, Hyderabad.
prasath2k6@gmail.com, drvr.sits@gmail.com, johnalexvlsi@gmail.com, sudhagarambur@gmail.com

Article Info
Volume 82
Page Number: 8057 - 8062
Publication Issue:
January-February 2020

Article History
Article Received*: 18 May 2019*
Revised*: 14 July 2019*
Accepted*: 22 December 2019*
Publication*: 05 February 2020*

Abstract
In recent years, Multi-Processor System-on-Chip (MPSoC) is the emerging trends in the field of electronic. In fundamental nature, the period of time is activated by this requirement to concentrate more difficult applications, in generally dipping the cost and power utilization of electronic devices. However, training those platforms cannot be done straight manner, mostly caused by complexity in terms of parallelization, moving the information from one place to other place and storage and in run-time resource management. Reduction of data access latency is the important key to achieve performance improvements in computing. For multiprocessors chip, the latency of data is mainly depending upon the memory hierarchy organization, on-chip interconnect, workload. More NoC designs are introduced to utilize the size of system to reduce latency, locality it is assigned by quick paths or circuits where the communication is faster than other path. The prototype signal are directly influenced by the cache group and a lot of cache groups are designed in separation of the multiple NoC or simple design of NoC, this maybe chance to missing the optimization techniques. The methodology of this work, co-design approach of NoC and cache group is introduced. The objective of new methodology is Nano caching system to interconnect with communication locality and thereby improving the system latency. It stores data that is primarily accessible by each processor in the locally accessible cache bank of the core and also operates dedicated high-speed circuits in interconnect to provide remote cores with rapid access to shared data.

Keywords: *NoC(Network on-Chip), Cache block, Data access latency.*

I. INTRODUCTION

Performance of the Muticore processor is a major concern in recent trends. It mainly depends on the latency of the data. The latency of the data is one of the main concerns for the design of on-chip interfaces and the organization of caches. The chip network affects the delay of the data stored in the cache memory and also the cache organization affects the distance between the chip storage space and the cores accessing the block. so that the communication standards over the NoC. It also affects the use of cache capacity which leads to a number of costly out-of-chip accesses. When the number of cores in the system increases and their by accessing the data delay is more.

The recent research to optimize the performance of the NoC and cache is only presence in the isolation of each other. For

January - February 2020
ISSN: 0193 - 4120 Page No. 01 - 11

example the optimization of the performance is achieved through reduction of global hop count, reducing the per hop latency through assigning fast paths or circuits. Static nonuniform cache architecture and private caches represent the two ends of the cache organization spectrum. Whatever neither of the perfect solution for CMPs. SNUC caches better utilize the cache capacity, but suffer from high data access latency since they interleave data blocks across physically distributed banks; thus rarely associating the data with the core that use it.

In contrast, private cache property, but suffer from a long delay of data access, as they insert blocks of data into naturally distributed banks. rarely associating data with the kernel that uses it. Conversely, private caches allow quick access to chip data blocks, but suffer from low cache capacity usage due to over-representation of data. Although simple NoC can be replaced by an improved one, we do not expect all cache organizations to equally benefit from an improved interface due to the access properties of each methodology. Here in this work, we take a co-design approach to the design of NoC and cache. We know only one more attempt at such codification. The NoC hybrid circuit is presented in conjunction with a SNUCA cache that uses an adaptation of the Cohesion Protocol specifically encoded with the NoC to promote and exploit the communication site. It targets the category of interfaces that exploit the communication site. the main purpose of the design process is to 1) improve cache access and utilize cache capacity 2) improve cache communication by reducing NoC traffic volume and promoting communication position.

Cache organization is the important issue in the data access latency, problem. In previous organization scheme, the private caches block access for the data block. If it is not present (if it made miss) within that block then it search from the off-chip memory. Then it stores in the cache block with Data Replacement policy. The term already told that the NoC design and Cache Organization is made with isolation between them. Now we proposed a new Nano cache scheme where the NoC design and cache organization is done as a codesign process. In the cache scheme process, three processes is done through it. One is Data placement and with look up table. The second one is Data Migration. The third one is Data Replacement Policy.

In the proposed new cache scheme that data placement and data migration is done at the same process. Then we made Data Replacement policy based on the proposed work. In the II section we discussed about the Data placement and look up table process. Then in the III section we have seen about the Data migration process. In the IV section the proposed Nano Cache scheme process is done. Fig.1 and Fig.2 and Chart I, II Explains the portrays about the new Data Replacement policy. IV section made the performance analysis report with Unique Private Caching Scheme and Nano Cache scheme. From Fig.3. The overall organization is explained for nano cache scheme.

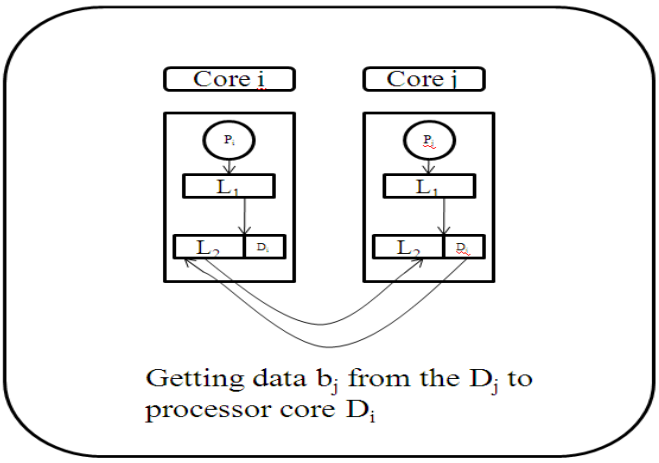


Fig 1.Cache block b_j is not on chip

January - February 2020
ISSN: 0193 - 4120 Page No. 01 - 11

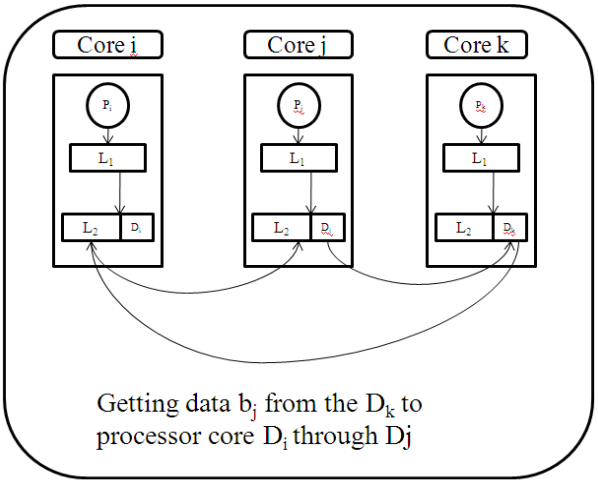


Fig 2. $L2_m$ is the host node of b_j

II.DATA PLACEMENT AND LOOK UP TABLE

In Chip Multi-processor, there are two cache blocks which are present. One is the private cache block L1 where the cache block is accessed by its own core processor. And the second cache block is L2 processor where the access is shared by the multi core-processor. From Fig.1.assume that the core P_i, where it has to retrieve the data b_j from the cache block. If the data is not in the private cache block it searches in the shared cache block $L2_i$ if the data is not in that block it made the data placement in the other core-processor. This is done of the NoC process. Here the L2 block uses the design process with Distributed Directory where it made the physical address of the home node and the host node of the each data in the L2 cache bank. So the searching in other core-processor is made with the directory bank. When the $L2_j$ doesn't have the data bj then it retrieve the data from the off-chip memory then it gives to the $L2_i$ block where it assumes that L2i is the host node and $L2_j$ is the home node in directory bank D_j. Suppose if the data bj is presented and used by other core-processor P_k then the directory of D_j made the assessment to the directory D_k. This process is shown in Figure.2. The directory D_k made the host node as $L2_i$ and it assigns the data to the $L1_i$ private cache block. Thus the Data placement where the required data is analyzed through the shared cache block other core processor by the Directory bank.

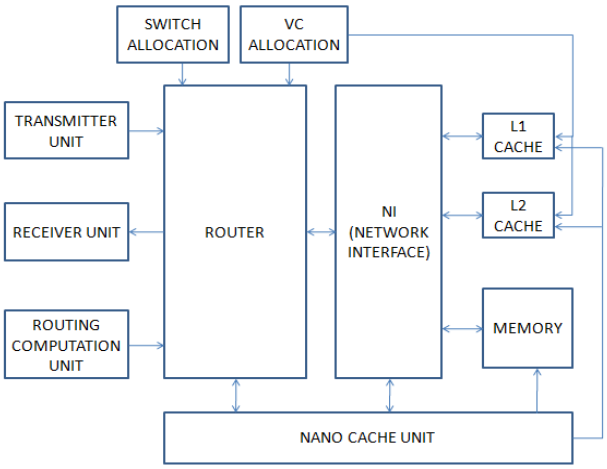


Fig 3. Full System Organization

III. DATA MIGRATION

After the data placement process is done by the processor, to make the data movement which is called as data migration from one core-processor to another is done. Formally it is done by the Gradual migration process, where the data migration is done whenever the data is need for the other core-processor. By doing the gradual data migration, some effects also produced. They are i) increased volume of traffic due to the gradual movement of the data units. ii) Reduced location of communication: Frequent movements may make it difficult for each tile to have a recognizable subset of other tiles showing most or all of the data exchange. The sharing user may experience increased block access time if they migrate. iii) Reduced effectiveness of local directory caches. In order to avoid that Unique private cache, uses the counter based Direct migration. Where the each core has the counter of the data to be accessed. Let us consider the data bj is accessed by the core Pi. When the data is required by share core Pj from Pi the data migration is done by the counter operation.

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 01 - 11

Whenever the data bj is accessed by the core Pi it makes the increment in the counter ci. And also whenever the data is accessed by Pj then counter cj made the count as incremented one. Thus calculating ci-cj=th then the data migration from Pi to Pj is done. Where the th is called as the threshold level of data migration values. But one important issue is it needs N counter for the core-processor. For that the proposed work make the use of only one counter in the home node processor. Whenever the data accessed by the P_i then counter made the reset value 0 in the counter. When the processor P_j is accessing the data b_j then counter c make the increment based on the directory bank. Data migration is done when the counter c=th. Thus we avoid the use of N counter in the processor and also get improved communication locality too.

IV.DATA REPLACEMENT POLICY

When the core processor accesses the data from off-chip memory, it is stored in the cache block. In order to store the data in cache block, the previous data block in the cache is to be replaced. It is done by the method called 'least recently used' policy. But it is not enough for the unique cache policy.The same thing is done in the nano cache scheme too. There is a difference in shared and non-shared block. Non-shared block means that the block can be only accessed by the local processor only. When we are considering the access process, the access time of private block is faster than the access time of the shared block. So normally the replacement policy is done by keeping the data in the private block as a same one and replacing only the shared block. But when the shared block of memory is replaced the access of data by other share processor make bottleneck when it called. So that we have to bias the least recently used policy. The accessing of shared block is more than the private block. So we proposed Biased LRU (SBLRU) to select the cache line to exit a synthetic set, S. Depending on an alpha parameter, if the number of private cache lines m within S satisfies m> = α, then the private ones are selected LRU cache lines to replace. If m <α, then the LRU cache line is replaced, regardless of whether it is shared or private. SBLRU can apply to any shared storage policy.

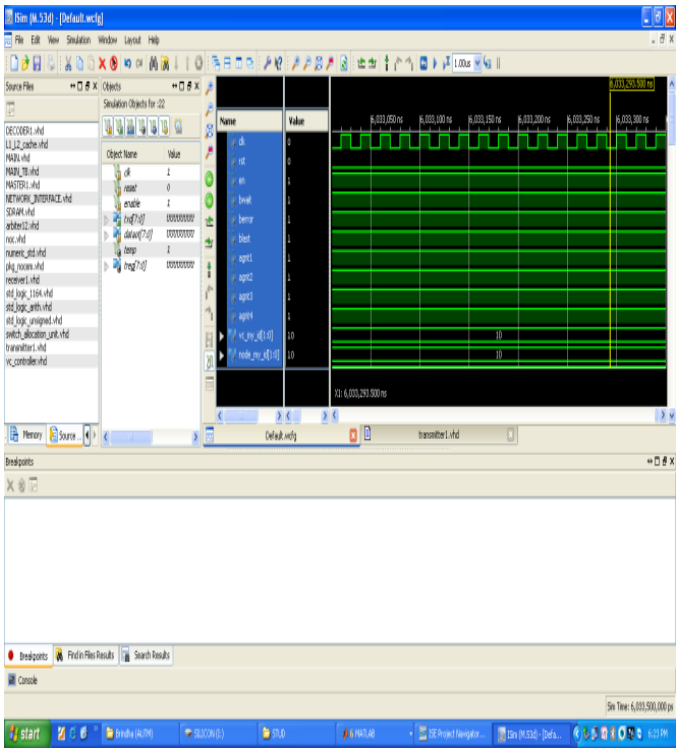


Figure 4. Output of Nano Caching Scheme

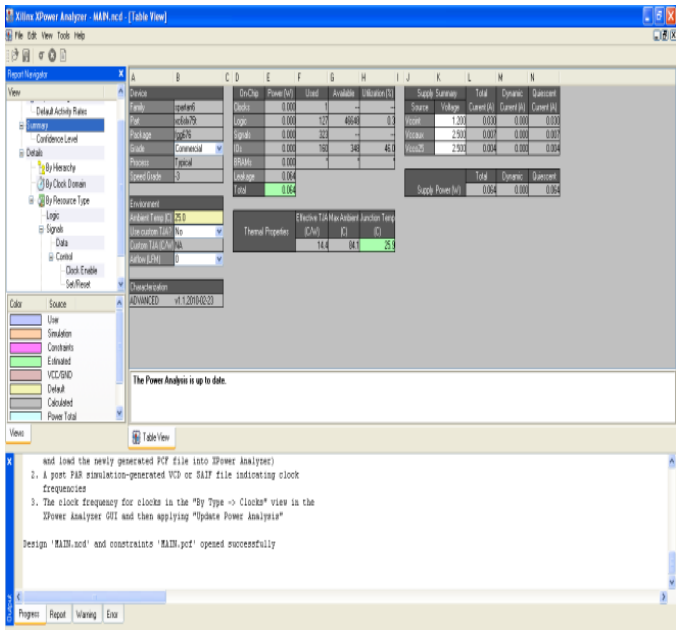


Figure 5. Power analysis of Nano Caching Scheme

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 01 - 11

The comparison of the existing and proposed system is given below,

CHART I. COMPARISON OF EXISTING & PROPOSED SYSTEM FOR AREA

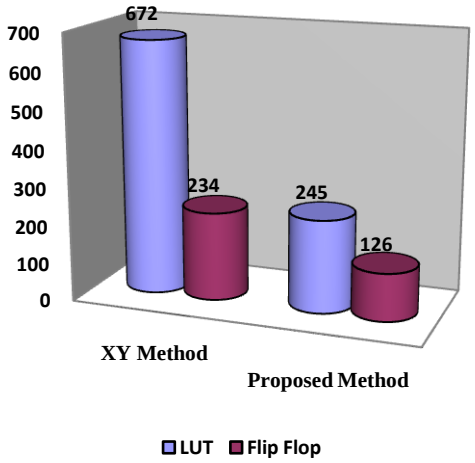
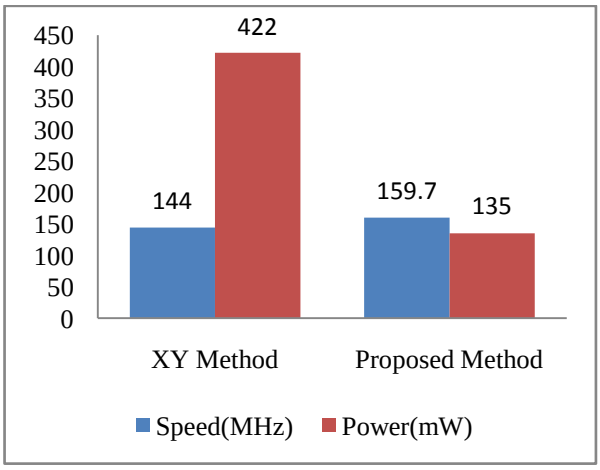


CHART 2. COMPARISON OF EXISTING & PROPOSED SYSTEM FOR SPEED& POWER



V. CONCLUSION

This article proposes Nano Cache Clustering; A novel distributed cache management scheme for a large-scale chip multiprocessor study reveals that: The global L2 chip cache can effectively relieve the memory pressure caused by computer thirsty data engines. However, its potential is still limited both by the bandwidth outside the chip and within the chip, especially as the number of active threads increases. Traffic congestion on the chip is largely due to intensive memory access requests issued from the cores in the chip. However, the application's runtime communication pattern is determined only by the design of the underlying memory hierarchy and the interconnection in the chip. These conclusions are generally applicable to a wide variety of multi-core processors with a similar design.

VI. REFERENCES

1. Abousamra, A.K.Jones, R.Melhem, "Codesign of NoC and Cache Organization for Reducing Access Latency in Chip Multiprocessors," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 6, pp. 1038-1045, 2012.
2. J. Kim, J.D. Balfour, and W.J. Dally, "Flattened Butterfly Topology for On-Chip Networks," Proc. IEEE/ACM 40th Ann. Int'l Symp. Microarchitecture (MICRO), pp. 172-182, 2007.
3. R.Arun Prasath**,** Dr. P.Ganeshkumar & E Sakthivel, "Performance Improvement of Hardware/Software Architecture for Real-Time Bio Application Using MPSoC", "Intelligent Automation & Soft Computing (Taylor & Francis)" Volume 23, Issue 02, pp. 351-357, October, 2016.
4. J.D. Balfour and W.J. Dally, "Design Tradeoffs for Tiled cmp On- Chip Networks," Proc. 20th Ann. Int'l Conf. Supercomputing (ICS), pp. 187-198, 2006.
5. S. Bourduas and Z. Zilic, "A Hybrid Ring/Mesh Interconnect for Network-on-Chip Using Hierarchical Rings for Global Routing," Proc. First Int'l Symp. Networks-on-Chip (NOCS), pp. 195-204, 2007.
6. R. Das, S. Eachempati, A.K. Mishra, N. Vijaykrishnan, and C.R. Das, "Design and Evaluation of a Hierarchical On-Chip Interconnect for Next-Generation CMPs," Proc. IEEE 15th Int'l Symp. High Performance Computer Architecture (HPCA), pp. 175-186, 2009.
7. Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang, "A Low-Radix and Low-Diameter 3D Interconnection Network Design,"Proc. IEEE 15th Int'l Symp. High Performance Computer Architecture(HPCA), pp. 30-42, 2009.
8. Grot, J. Hestness, S.W. Keckler, and O. Mutlu, "Express Cube Topologies for On-Chip Interconnects," Proc. IEEE 15th Int'l Symp. High Performance Computer Architecture (HPCA), pp. 163-174, 2009.
9. Kumar, L.-S. Peh, P. Kundu, and N.K. Jha, "Express Virtual Channels: Towards the Ideal

January - February 2020
ISSN: 0193 - 4120 Page No. 01 - 11

Interconnection Fabric," Proc. 34th Ann. Int'l Symp. Computer Architecture (ISCA), pp. 150-161, 2007.
10. N.D.E. Jerger, L.-S. Peh, and M.H. Lipasti, "Circuit-Switched Coherence," Proc. ACM/IEEE Second Int'l Symp. Networks-on-Chip (NOCS), pp. 193-202, 2008.
11. E. Bolotin, Z. Guz, I. Cidon, R. Ginosar, and A. Kolodny, "The Power of Priority: NoC Based Distributed Cache Coherency,"Proc. First Int'l Symp. Networks-on-Chip (NOCS), pp. 117-126, 2007.
12. S. Shao, A.K. Jones, and R. Melhem, "Compiler Techniques for Efficient Communications in Circuit Switched Networks for Multiprocessor Systems," IEEE Trans. Parallel and Distributed Systems, vol. 20, no. 3, pp. 331-345, Mar. 2009.
13. Y. Li, A. Abousamra, R. Melhem, and A.K. Jones, "Compiler-Assisted Data Distribution for Chip Multiprocessors," Proc. 19th Int'l Conf. Parallel Architectures and Compilation Techniques (PACT),2010.
14. M.K. Qureshi, "Adaptive Spill-Receive for Robust High-PerformanceCaching in CMPs," Proc. IEEE 15th Int'l Symp. HighPerformance Computer Architecture (HPCA), pp. 45-54, 2009.
15. X. Wang, M. Yang, Y. Jiang, P. Liu, Power-aware mapping for Network-on-Chip architectures under bandwidth and latency constraints, in: International Conference on Embedded and Multimedia Computing (EM-COM), Jeju, Korea, 2009, pp. 1–6.
16. E. Carvalho, C. Marcon, N. Calazans, F. Moraes, Evaluation of static and dynamic task mapping algorithms in NoC-based MPSoCs, in: International Symposium on System-on-Chip, 2009, SOC, Tampere, Finland, 2009, pp. 87–90.
17. R.Arun Prasath, Dr. P.Ganeshkumar, "Design of Low Power Level Shifter Circuit with Sleep Transistor Using MultiSupply Voltage Scheme", "Circuits and Systems" Volume 07, Issue 07, pp. 1132- 1139, ISSN: 2153-1293, 2016.