Overcoming and Analyzing the Bottleneck of Interposer Network in 2.5D NoC Architecture

Chen Li, Zicong Wang, Lu Wang, Sheng Ma, and Yang Guo(✉)

College of Computer, National University of Defense Technology,
Changsha 410073, China
{lichen, wangzicong, luwang, masheng,
guoyang}@nudt.edu.cn

Abstract. As there are still a lot of challenges on 3D stacking technology, 2.5D stacking technology seems to have better application prospects. With the silicon interposer, the 2.5D stacking can improve the bandwidth and capacity of memory. Moreover, the interposer can be explored to make use of unused routing resources and generates an additional network for communication. In this paper, we conclude that using concentrated Mesh as the topology of the interposer network faces the bottleneck of edge portion, while using Double-Butterfly can overcome this bottleneck. We analyze the reasons that pose the bottleneck, compare impacts of different topologies on bottlenecks and propose design goals for the interposer network.

Keywords: 2.5D stacking technology · Topology · Interposer network · Performance bottleneck

1 Introduction

Recently, process scaling becomes increasingly difficult to maintain Moore's law. Some technologies emerge to continuously develop the semiconductor integrated circuit, such as multi-core, multi-threading and virtualization technologies. However, these technologies face the challenges of the Memory Wall [1]. Therefore, the three-dimensional (3D) stacking technology has emerged to deal with these problems, as it offers interconnect length reductions, memory bandwidth improvements, heterogeneous integration and smaller chip sizes.

Although 3D stacking technology has many benefits to the conventional 2D layout, there are several challenges that could potentially hinder its adoption, such as the thermal issue, the absence of EDA tools and testing issues [2]. In comparison, silicon interposer-based stacking, known as "2.5D stacking" [3], is gaining more traction [4]. As shown in Fig. 1, with 2.5D stacking technology multiple silicon dies can be stacked side-by-side on a silicon interposer carrier. The 3D-stacked approach is a revolutionary approach that it needs new co-design and methods for design flow and testing, while the 2.5D-stacked approach is evolutionary [5]. It side-steps many challenges in 3D stacking and has been supported by current design tools.

© Springer Science+Business Media Singapore 2016
J. Wu and L. Li (Eds.): ACA 2016, CCIS 626, pp. 40–47, 2016.
DOI: 10.1007/978-981-10-2209-8_4

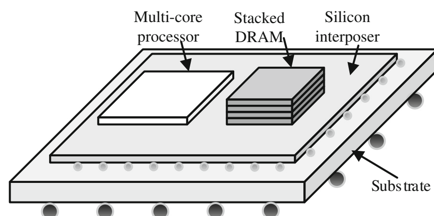


Fig. 1. 2.5D stacking technology

Recent years, some commercial 2.5D-stacked products have already emerged [6, 7]. The most widely application of 2.5D stacking technology is the integration of memory (DRAM) with a multi-core processor. Larger capacities and higher bandwidth for in-package memory can be offered by the silicon interposer, as it has enough areas for much memory to be integrated and many thousands of connections available across the interposer. The interposer memory stacking also requires large bandwidth for processor-to-memory traffic. In order to continuously increase the bandwidth, previous work [8, 9] shows that significant routing resources inside the silicon interposer can be exploited to implement an additional network. We call it interposer network in this paper.

The topology determines the physical layout and connections between nodes and channels in the network. Moreover, the number and locations of TSVs depend on the topology. It is thus clear that the effect of a topology on overall network cost-performance is profound. There are many topologies can be implemented in the interposer network. Owing to the simplicity and scalability, the Mesh has been widely used in CMPs [10, 11]. In order to reduce the μbump area overhead, the concentrated method is used that four nodes in CPU multi-core layer connects one node in the interposer network.

In this paper, we conclude that using the concentrated Mesh as the topology of the interposer network faces the bottleneck of edge portion network, while using Double-Butterfly can overcome this bottleneck. We analyze the reasons that pose the bottleneck, compare impacts of different topologies on bottlenecks and propose design goals for the interposer network.

2 Target System and Evaluation Methodology

In our 2.5D interposer-based system, a 64-core CPU and 4 stacked DRAMs are stacked on a silicon interposer [8]. In order to reduce the cost of NoC in the interposer (TSV/μbump) [12], the topology of our 2.5D NoC architecture is Mesh on the CPU die and Concentrated Mesh or Double-Butterfly on the interposer die shown in Fig. 2. The concentrated method means that each of the 16 interposer nodes connects four nodes on the CPU die. There are totally 8 nodes of memory controllers on left and right sides of the interposer network. Each one connects a nearby interposer node. Figure 2 also shows two types of interposer implementations. In the near term, passive type without active devices in the interposer is a practical way, while active interposer is more likely to be a 3D integrated way. That is to say all logic/gates are placed on the CPU die and

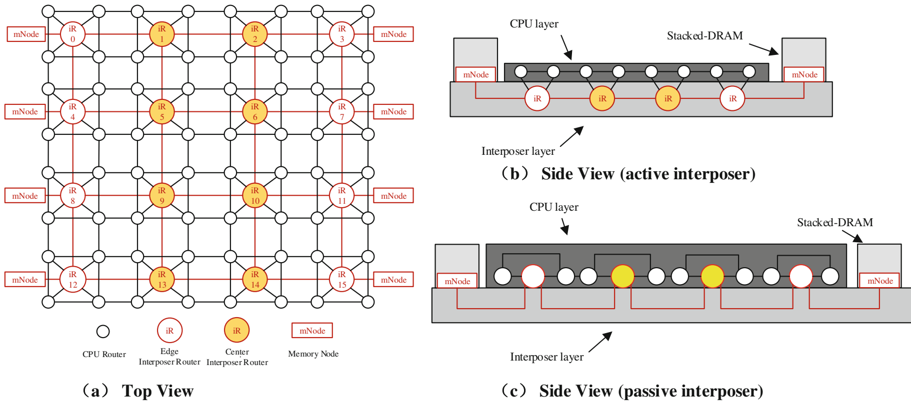


Fig. 2. Target system (Color figure online)

only metal routing on the passive interposer. Besides, there are two types of traffic, including the core-to-core coherence traffic transferred on the CPU die and the core-to-memory traffic transferred on the interposer die.

We use a cycle accurate interconnection network simulator (Booksim) [13] for the evaluation. We modify Booksim to implement our 2.5D NoC architecture. As the comparison will be focused on the interposer layer topologies, all configurations use an 8×8 Mesh for the multi-core die. We evaluated the CMesh, CMesh2 and DB (Double-Butterfly) topologies on interposer layer as shown in Fig. 3. Our NoC designs utilize 4

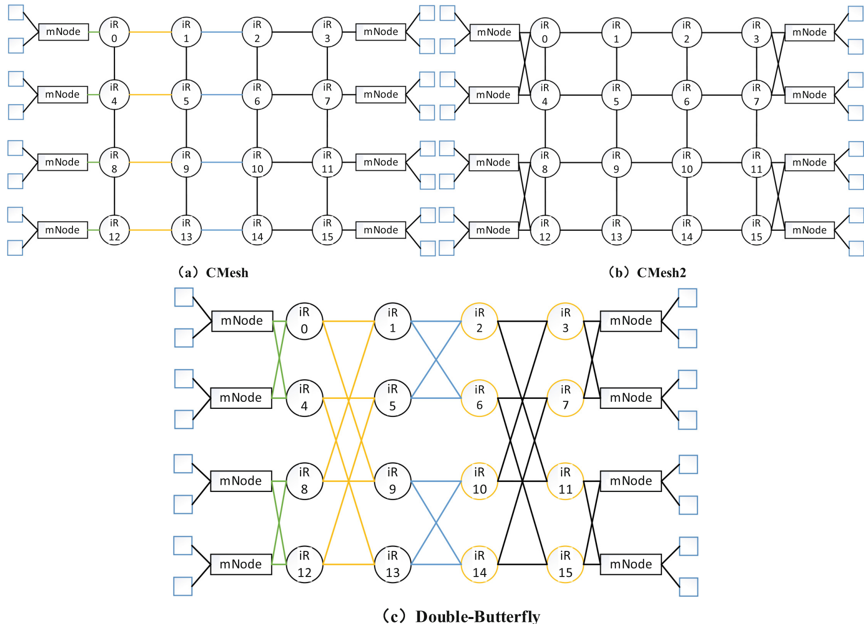


Fig. 3. Topologies

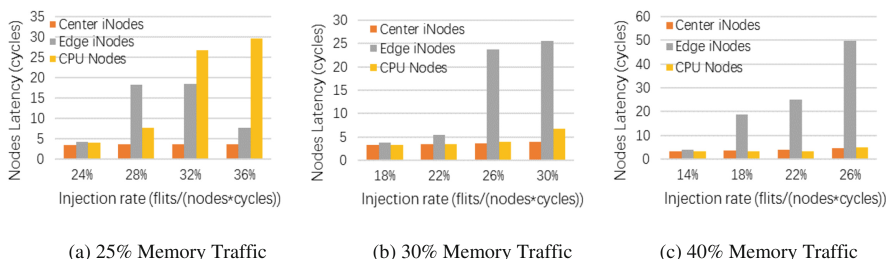
cycles router and 2 cycles link for the interposer layer. There are totally four DRAM stacks on the interposer. Each DRAM stack provides four memory channels, for a system-wide total of 16 channels. Each two channels share a memory node. The interposer layer network dimensions include 8 memory nodes that interface with the DRAM stacks memory channels.

3 Bottleneck Description and Analysis

3.1 Bottleneck Description

As the 2.5D NoC architecture leverages Mesh on the CPU die, CMesh on the interposer die and memory nodes are located in two sides, the topology of the whole 2.5D NoC is asymmetric. There are 3 possible performance bottlenecks of the 2.5D NoC architecture, including the upper layer network (Black nodes), the center portion of the lower layer (Yellow nodes) and the edge portion of the lower layer (Red nodes), as shown in Fig. 2(a). Any one of these parts may lead the 2.5D NoC to be saturated, while other partial networks are still working in unsaturated state.

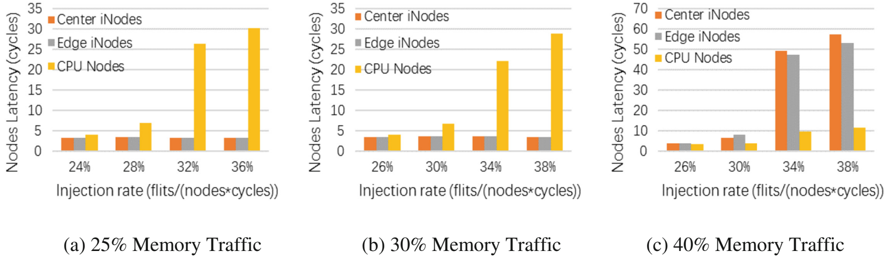
We evaluate average latencies of messages passing through network nodes in these 3 parts. We leverage the baseline design with XY-Z routing, and results are shown in Fig. 4. We find that CPU nodes on the upper layer lead the whole network saturation when memory traffic accounts for 25 % of total traffic. The bisection bandwidth of the upper layer is two times of the bisection bandwidth of the lower layer. Thus, when memory traffic occupancy rate is more than 30 %, the lower layer becomes the bottleneck. Figure 4 shows that edge nodes on the lower layer lead the whole network saturation when the percentage of memory traffic is larger than 30 %. However, when edge interposer nodes are saturated, latencies of messages passing through center interposer nodes are still low. Even when the memory traffic account for larger than 50 % of total traffic, the edge network of the lower layer is still the performance bottleneck.



(a) 25% Memory Traffic (b) 30% Memory Traffic (c) 40% Memory Traffic

Fig. 4. Performance bottlenecks of CMesh

First, we suppose that the bottleneck of edge portion comes from the small bandwidth of edge portion. We evaluate the CMesh2 with more bandwidth in the edge portion as shown in Fig. 3(b). Compared with CMesh, we add 4 links on each side of edge portion network. However, the evaluation result shows that the edge portions are



(a) 25% Memory Traffic (b) 30% Memory Traffic (c) 40% Memory Traffic

Fig. 5. Performance bottlenecks of double-butterfly

still the performance bottleneck of the whole network. Then, we find that DB overcomes the bottleneck of edge portion. Workloads in interposer network are balanced and uniform as shown in Fig. 5(c). In next subsection, we will compare these two topologies in the interposer network, and then analyze reasons that pose the bottleneck. Based on the analysis, we will propose design goals of the interposer network in 2.5D interposer-based system.

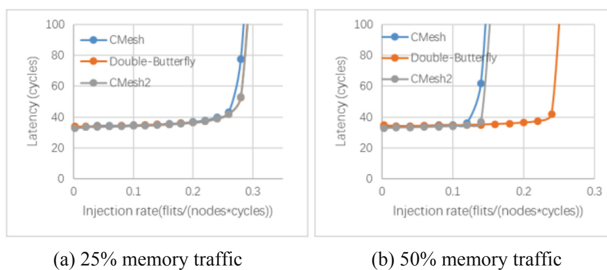
3.2 Impacts of Topologies on Bottlenecks

We compare these two topologies and analyze their features in following points: hops of memory traffic, link utilization, path diversity, bisection bandwidth and Latency-Injection rate. We can find out impacts of topologies on bottlenecks.

1. Hops of memory traffic
   The interposer network mainly used for transferring memory traffic. Those messages are injected from nodes in multi-core layer to memory nodes on edge sides of interposer layer. Thus, the average hops of memory traffic are 6 on CMesh and 4.75 on DB according to our computing. The experimental result of average hops in uniform pattern are 5.7 on CMesh and 4.7 on DB. Obviously, DB has lower average hops compared with CMesh.
2. Single hop latency and zero load latency
   The pipeline latency of router is 4 cycles. The link latency of CMesh is 2 cycles. The link latency of DB is 2/4/6 (average 3.4 cycles) for links with different length. Although the average hops of CMesh is larger than DB, their zero load latencies are nearly the same. That is because the link latency of DB is longer than CMesh, and lower hops amortize the longer link latency.
3. Link utilization
   We compare the link utilization of both topologies in uniform pattern. As shown in Fig. 3(a), considering different portions of links for CMesh, the utilization ratio of blue links is 25 %, while the yellow links is 37.5 % and the green links is 50 %. The other side is symmetrical with this side.
   The link utilization of DB is similar to CMesh. For DB, the utilization ratio of blue links and green links are the same with CMesh. The utilization of yellow links is 43.75 %. The 6.25 % more utilization comes from the case that message from IR0

or IR4 need to be transferred to the lower half of the memory channels on the left side. In this case, routes must divert to the previous stage. We can find that the link utilization of both topologies are similar and it has little impact on the bottleneck.

4. Path diversity
   The XY-Z routing is leveraged in CMesh. As the deterministic routing is leveraged, there is no path diversity in CMesh. The path diversity of DB is a little complex. The path diversity of memory traffic which need to be transferred through blue links are 2, while the path diversity is 1 in other situation.
   In some traffic patterns, such as hotspot, no path diversity may make some links fall into high traffic pressure. If packets are from yellow nodes to the left memory nodes in DB as shown in Fig. 3(c), it can choose a path with low workload to the destination node. Thus congestion can be alleviated.
5. Bisection bandwidth
   For CMesh, the ratio of bisection bandwidth between the upper layer and the lower layer is 2:1 (8:4). For DB, the ratio of bisection bandwidth between the upper layer and the lower layer is 1:1. Only nodes at edge sides can consume packets, while center nodes are just used as switch.
   It answers the reason why the lower layer of network becomes the bottleneck when the percentage of memory traffic is larger than 30 % for CMesh and 40 % for DB.
6. Latency-Injection rate
   Figure 6 shows the performance comparison between CMesh, DB and CMesh2 in uniform traffic pattern. As shown in Fig. 6(a), when the memory traffic makes up 25 % of the total traffic, their performance are nearly the same. This is because the saturation of all three topologies are caused by the saturation of CPU layer network in 25 % memory traffic.
   When the memory traffic accounts for 50 % of the total traffic, the performance of CMesh and CMesh2 are nearly the same, while the average performance gain of DB over CMesh is 54.5 %. Considering the performance bottleneck in high memory traffic, we can find that the performance gain of DB comes from overcoming the bottleneck of the edge portion network. A uniform and balanced network performs high efficiently. CMesh2 does not overcome the bottleneck of edge network. It shows that adding bandwidth in edge side is useless.



(a) 25% memory traffic (b) 50% memory traffic

Fig. 6. Performance comparison

3.3 Summary and Design Goals of Interposer Network

Based on the comparison between CMesh and DB, we know that the link utilization of them are similar, link utilization is not the key factor of the performance bottleneck. Both single hop latency and zero load latency also have little impact on the bottleneck. In fact, the interposer network is similar to GPGPU network that the performance is more sensitive to interconnect bisection bandwidth rather than latency [14, 15]. On the contrary, the bisection bandwidth and average hops pose strong impact on the bottleneck. Larger bisection bandwidth makes larger throughput. Lower average hops reduce contention of the interposer network. Furthermore, compared with CMesh, DB improves the path diversity giving more routing choices to messages when being transferred. It leads the load to be balanced. When the workload increases in the interposer network, the high throughput highlights the advantage of lower contention and high bandwidth.

Therefore, we can conclude some design goals of the interposer network based on topologies analysis. First, in order to reduce the contention, we should try best to reduce average hops between the source and destination nodes. Leveraging long metal wires is a suitable way in interposer layer network, due to its abundant metal routing resource. Second, higher throughput needs higher bisection bandwidth. We should improve the bisection bandwidth through making connections between nodes as many as possible. Third, the interposer network should provide the path diversity as much as possible. As all nodes except memory nodes in the interposer network are switches, they are just used for transferring packets and cannot absorb packets. Thus, deterministic routing algorithms are not as suitable as minimal adaptive routing algorithms which provide more path diversity. It can balance the workload on the interposer network.

4 Conclusion

The 2.5D stacking technology leverages an interposer to stack chips and DRAMs. Making use of the metal layer on the interposer provides fascinating opportunities to explore new features on 2.5D NoC architecture. In this paper, first we find that the edge portion of interposer network in CMesh always lead the saturation of the whole 2.5D network when the memory traffic is larger than 30 % of the total traffic. We compare it with CMesh2 and DB. DB can overcome this performance bottleneck. Then we analyze their features and find out reasons that pose this performance bottleneck. At last, we propose design goals of the interposer network.

In the future, we will focus on the interposer layer network. On one hand, exploit the design space of interposer layer network; on the other hand, design a high efficient interposer network for the reply network of GPGPU-Memory 2.5D system.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (No.6133007, No. 61303065), Doctoral Fund of Ministry of Education (20134307120028).

References

1. Wulf, W.A., McKee, S.A.: Hitting the memory wall: implications of the obvious. ACM SIGARCH Comput. Archit. News **23**(1), 20–24 (1995)
2. Xie, J., Zhao, J., Dong, X., Xie, Y.: Architectural benefits and design challenges for three-dimensional integrated circuits. In: 2010 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 540–543, December 2010
3. Deng, Y., Maly, W.P.: Interconnect characteristics of 2.5-d system integration scheme. In: Proceedings of the 2001 International Symposium on Physical design, pp. 171–175. ACM (2001)
4. Loh, G.H., Jerger, N.E., Kannan, A., Eckert, Y.: Interconnect memory challenges for multi-chip, silicon interposer systems. In: Proceedings of the 2015 International Symposium on Memory Systems, pp. 3–10. ACM (2015)
5. Bolsens, I., Xilinx, C.: 2.5D ICs: Just a stepping stone or a long term alternative to 3d? In: Keynote Talk at 3-D Architectures for Semiconductor Integration and Packaging Conference (2011)
6. AMD: Amd radeon r9 fury x graphics card (2015). http://support.amd.com/documents
7. Saban, K.: Xilinx stacked silicon interconnect technology delivers breakthrough FPGA capacity, bandwidth, and power efficiency. Xilinx White paper: Vertex-7 FPGAs (2011)
8. Jerger, N.E., Kannan, A., Li, Z., Loh, G.H.: Noc architectures for silicon interposer systems: Why pay for more wires when you can get them (from your interposer) for free? In: 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 458–470. IEEE (2014)
9. Kannan, A., Jerger, N.E., Loh, G.H.: Enabling interposer-based disintegration of multi-core processors. In: Proceedings of the 48th International Symposium on Microarchitecture, pp. 546–558. ACM (2015)
10. Howard, J., Dighe, S., Hoskote, Y., et al.: A 48-core IA-32 message-passing processor with DVFS in 45 nm CMOS. In: 2010 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pp. 108–109. IEEE (2010)
11. Wentzlaff, D., Griffin, P., Hoffmann, H., et al.: On-chip interconnection architecture of the tile processor. IEEE Micro **5**, 15–31 (2007)
12. Liu, C., Zhang, L., Han, Y., Li, X.: Vertical interconnects squeezing in symmetric 3d mesh network-on-chip. In: Proceedings of the 16th Asia and South Pacific Design Automation Conference, pp. 357–362. IEEE Press (2011)
13. Jiang, N., Becker, D.U., Michelogiannakis, G., Balfour, J., Towles, B., Shaw, D.E., Kim, J., Dally, W.J.: A detailed and flexible cycle-accurate network-on-chip simulator. In: 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 86–96. IEEE (2013)
14. Bakhoda, A., Kim, J., Aamodt, T.M.: Throughput-effective on-chip networks for manycore accelerators. In: Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture, pp. 421–432. IEEE Computer Society (2010)
15. Bakhoda, A., Yuan, G.L., Fung, W.W., Wong, H., Aamodt, T.M.: Analyzing CUDA workloads using a detailed GPU simulator. In: Proceedings of the International Symposium on Performance Analysis of Systems and Software, April 2009