Design and Early Evaluation of a 3-D Die Stacked Chip Multi-Vector Processor

Ryusuke Egawa*§, Yuseke Funaya†‡, Ryu-ichi Nagaoka‡, Akihiro Musa‡, Hiroyuki Takizawa‡§ and Hiroaki Kobayashi*§

*Research Division on Supercomputing Systems, Cyberscience Center, Tohoku University, Sendai, 980-8578, Japan
Email: [egawa, koba]@isc.tohoku.ac.jp
†GSIS, Tohoku University University, Sendai, 980-8578, Japan
Email: [funaya.sc, nagaoka.sc, tacky]@isc.tohoku.ac.jp
‡NEC Corporation, 108-8011, Tokyo, Japan, Email: a-musa@bq.jp.nec.com
§JST CREST
*Research Fellow of the Japan Society for the Promotion of Science

Abstract—
Modern vector processors have significant advantages over commodity-based scalar processors for memory-intensive scientific applications. However, vector processors still keep single core architecture, though chip multiprocessors (CMPs) have become the mainstream in recent processor architectures. To realize more efficient and powerful computations on a vector processor, this paper proposes a 3-D stacked chip multi-vector processor (CMVP) by combining a chip multi-vector processor architecture and the coarse-grain die stacking technology. The 3-D stacked CMVP consists of I/O layers, core layers and the vector cache layers. The I/O layer significantly improves off-chip memory bandwidth, and the vector core layer enables to install many vector cores on a die. The vector cache layer increases the capacity of on-chip memory and a high memory bandwidth to achieve the performance improvement and energy reduction by decreasing the number of off-chip memory accesses. The results of performance evaluation using real scientific and engineering applications show the potential of the 3-D stacked CMVP. Moreover, this paper clarifies that introducing the vector cache is more energy-effective than increasing the off-chip memory bandwidth to achieve the same sustained performance on the 3-D stacked CMVP.

I. INTRODUCTION

Modern vector processors have significant advantages over commodity-based scalar processors for memory-intensive scientific applications[1]. However, vector processors still keep a single core architecture, though chip multiprocessors (CMPs) have become the mainstream in recent processor architectures. To realize more efficient and powerful computations on a vector processor, CMP architectures should be applied to vector processor design in the near future. Since the computational efficiency of vector processors relies on their high memory bandwidth, a novel memory design that provides each vector core on a chip with a sufficiently-high memory bandwidth is strongly required to keep the high computational efficiency on a chip multi-vector processor (CMVP).

So far, an architectural design of CMVP with a shared cache memory has been proposed and its potential has been clarified [2]. However, this article has not discussed its feasibility and implementations. Essentially, vector processors need a plenty of hardware resources compared to scalar processors. This is because a vector processor requires many vector pipelines, large vector registers and vast I/O logic to simultaneously process a huge amount of data provided at a high-memory bandwidth[3]. In addition, CMVP needs a large shared on-chip memory named a vector cache to keep a high sustained memory bandwidth. Therefore, even if technology scaling were to advance as ever, it would be difficult to implement the CMVP with many vector cores by the conventional 2-D implementation technology due to the area limitation.

To realize CMVPs, one possible way is to adopt 3-D integration technologies, which can significantly improve an integration density of CMOS transistors by stacking silicon dies. Though 3-D integration technologies are not brand new, recently 3-D integration technologies have come under the spotlight to overcome the limitations of conventional 2-D microprocessor implementation. Due to the appearance of a through-silicon-via (TSV) with high feasibility and density[4], computer architects and circuit designers are re-attracted to 3-D integration technologies to expand the design space of future microprocessors. There are a lot of benefits by introducing 3-D integration technologies, such as an increase in integration density, wire length reduction, power reduction with a low area overhead. To fully exploit the benefits from the 3-D stacking with TSVs, coarse-grain die stacking of memories and cores with TSVs is expected as the most promising approach to the memory wall problem[5],[6],[7].

Under this situation, this paper proposes a 3-D stacked chip multi-vector processor by combining the CMVP architecture and the coarse-grain die stacking technology. Although 3-D integration technology brings us a huge amount of silicon budget, so far an effective way to use a plenty of resources has not been discussed from the viewpoint of architecting microprocessors. Therefore this paper also tries to clarify the effective 3-D die-stacking and resource usage by evaluating
several configurations of the 3-D stacked CMVP. This paper also describes early performance evaluation of the 3-D stacked CMVP using real science and engineering application codes.

The rest of this paper is organized as follows. Section 2 presents the background of the 3-D integration technology, and reviews related work. Section 3 introduces basic concepts of the CMVP architecture, and Section 4 presents a 3-D stacked CMVP and its implementations. Early performance evaluation of the 3-D stacked CMVP is carried out in Section 5 to demonstrate its potential. Section 6 concludes this paper.

II. 3-D INTEGRATION TECHNOLOGIES

A. A Vertical Interconnect for 3-D Integration

Since device engineers have been able to steadily decrease the feature size, it improves an integration degree and the performance of transistors. So far, the technology scaling brings us higher performance and higher levels of on-chip functional integration. However, the rapid increase in speed and complexity of recent microprocessors is outstripping the benefit of feature size scaling. Besides, this continuous progress causes an increase in wire delay, power density and design complexity, then it is getting harder to design high-performance microprocessors by the conventional 2-D implementation. However, computer architects try to design a high performance and functionality microprocessor with a high device density to response ever-increasing demands of application engineers.

3-D integration is an emerging fabrication technology that vertically stacks plural integrated chips. The benefits include an increase in device density, flexibility of routing and the ability to integrate disparate technologies[8][9]. Although there are a lot of technologies such as wire bonding to realize vertical stacking of two or more integrated circuits, this study focuses on a 3-D integrated circuit with vertical interconnects named TSVs due to their small RC delay with a high density. Many researchers have reported processing technology for thin and long TSVs with high feasibility[10][11][12]. For example, the current fabrication technology can achieve 1,000,000 TSVs per cm²[13].

Two topologies can be conceived to bind two silicon dies: face-to-face and face-to-back, where face is the side with the metal layer and back is the side with the silicon substrate as shown in Figure 1. In the face-to-face bonding, die-to-die (D2D) vias or micro bumps are processed and deposited on top of metal layers as the conventional metal etching technologies. Although the face-to-face bonding can provide a higher D2D via density and a lower area overhead than face-to-back, it can just allow to stack only two active silicon layers. On the other hand, the face to back bonding can stack any number of multiple active silicon layers by TSVs that go through silicon bulk with lower via density. As noted earlier, the dimension for TSVs varies from 2 to 5μm in recent real implementations, and the pitch of 3-D vias only requires a modest overhead. Koyanagi et.al have reported that recent high-density TSV process technologies can form tens of thousands of TSVs in a chip with an area penalty of 1mm²[4].

More detailed descriptions of TSV implementation techniques are shown in [4][10]. To realize aggressive stacking of multiple layers, face-to-back seems preferable. In this paper, we focus on the face-to-back stacking with TSVs to realize aggressive stacking of cores and shared memories for fully exploiting the potential of CMVP.

B. Related Work

Several studies in the literature have explored a new architectural design space based on 3-D integrated technologies. The benefits obtained by applying the 3-D integration technology to the memory-subsystem design are considered to realize a larger on-chip memory capacity, a shorter memory access latency, a higher memory bandwidth, and a lower energy consumption. Due to these effects, computer architects have focused on the high-speed feature of TSVs to overcome the memory wall problem, and hence stacking a processor and a memory sub-system on a 3-D integrated chip [6][7][14].

According to the trend of recent microprocessor designs, several researches have combined 3-D memory stacking and CMPs to supply data to a chip with a massive number of cores at an enough on-chip memory bandwidth[15][16]. Black et al. have explored the memory bandwidth benefits using Intel Core2 Duo processor [5]. By stacking an additional cache memory layer, the on-chip cache capacity is increased, and the performance is improved by reducing off-chip memory accesses. Loh et al. have discussed DRAM stacking on CMPs[17]. They have tried to fully take advantages of the benefit of 3-D integration technologies with TSVs, and memory organizations are optimized for many purposes such as the main memory or the last level caches. In addition, stacking layers of non-volatile cache memory [18][19] are also studied to mitigate the effects of the processor-memory speed gap. However, these researches are carried out based on multi-core or many-core scalar microprocessors.

In this paper, we focus on the chip multi-vector processor, which cannot be implemented by conventional 2-D technologies with a high memory bandwidth. Furthermore, since the performance of the vector architecture strongly depends on a high memory bandwidth compared to the scalar architecture, the 3-D integration technology is suitable for the vector architecture.

III. CMVP : A CHIP-MULTI VECTOR PROCESSOR

Recently, CMPs have become the main stream in commodity scalar processors. Eight-cores CMPs are already in the
commercial market, and an eighty-cores CMP is prototyped by Intel to overcome power and performance limitations of single-core architectures [20]. On the other hand, CMP-based vector processors have not been found as real implementations. However, the CMP architecture is also promising for vector processor design, because recent scientific and engineering applications running on a vector supercomputer are well parallelized by vector compilers and/or OpenMP. The computational granularity of the multi-threads is coarser than the granularity of loop vectorization. Therefore, a CMP-based vector architecture that can achieve hierarchical parallel processing is promising for future vector processor design to exploit various levels of parallelism. Under this situation, CMVP architecture has been proposed by Musa et al. [2].

For CMVP, it is getting harder to further improve the performance and energy efficiency due to several limitations such as the die area and the number of I/O pins on a chip. The most severe problem is the decrease in the ratio of memory bandwidth to the floating-point operation rate (Bytes/Flop, B/F). A high B/F ratio is essential to achieve a high computational efficiency, i.e., efficient use of the computing power. If the B/F ratio of a vector processor decreases to be as the same level as that of a scalar one, the vector processor will no longer be able to keep its superiority over the scalar processor in terms of the sustained performance.

Under this situation, to compensate for a decreasing B/F ratio of recent vector processors, an on-chip memory for a vector processor has been proposed [21]. The on-chip memory can provide data to vector registers of the processor at a high bandwidth because the data transfer is not limited by I/O pins. Hence, the B/F ratio of a vector processor is improved by storing reusable data in the on-chip memory to achieve high sustained performance. In addition, an on-chip memory decreases the number of off-chip main memory accesses, resulting in decreasing the energy consumption in the processor I/O, memory network and off-chip memory components. Therefore, the vector cache is also one of key components in CMVP design.

The vector cache is not private to each processor core but shared by multiple cores because scientific simulations such as difference schemes often have a high locality of memory reference among threads. The vector cache consists of 32 two-way set-associative sub-caches with miss status handling registers (MSHR) [22]. A vector load/store instruction of the vector architecture is able to concurrently deal with up to 256 floating-point data. Hence, the vector cache also needs to process up to 256 data in continuity. Furthermore, the vector cache employs a bypass mechanism between the main memory and vector register files. The bypass mechanism makes possible to supply data from both the main memory and the vector cache at the same time. Thus, the total amount of data provided to the vector register files in time is increased by the bypass mechanism.

In addition, the vector cache employs the MSHR mechanism to realize a non-blocking cache. In scientific computations such as difference schemes, two vector load instructions load the memory regions that are partially overlapped. If the subsequent load instruction is issued right after the preceding instruction, however, the data to be fetched by the preceding instruction have not been cached in the vector cache yet owing to the long latency of main memory accesses. Thus, the subsequent load instruction causes cache misses even though the data to be accessed are in-flight. To avoid this situation, MSHR holds information of in-flight and subsequent load requests. When the memory address of a subsequent load request is same as the memory address of an in-flight load data, the subsequent load request is not sent to the main memory. Then, the subsequent load requests wait an arrival of the in-flight load data at MSHR. It make possible the subsequent load instruction to reuse in-flight data fetched by the preceding instruction. As a result, MSHR can prevent redundant data transfer from the main memory, resulting in efficient use of the limited memory bandwidth.

Figure 2 shows the CMVP block diagram, which has several vector cores and a shared vector cache. Each core has 32 memory ports. The cores and the vector cache are interconnected through 32 crossbar switches. The off-chip main memory employs an interleaved memory for a high memory bandwidth. Therefore, the vector cache also consists of multiple blocks, each connecting to an off-chip memory bank. These blocks will be referred to as sub-caches. Each crossbar switch has a priority control mechanism of data transfer from the cores to the vector cache. When two or more cores send data at once, the data are forwarded to the vector cache according to their priorities. In this paper, a higher priority is given to a smaller-numbered core.

IV. DESIGNING 3-D STACKED CMVP

In this section, we discuss the design and implementation of a 3-D stacked CMVP. Considering the implementation of 3-D integrated circuits (3DICs) with TSVs, each layer should be fabricated at a high yield rate because the yield of each layer is a key factor to keep the cost low of 3-DICs low. In addition, it has been reported that 3-D die-stacking with stable CMOS technology improves the cost of 3DICs [23]. Therefore, we assume the SX-8 vector processor, which was implemented in 90nm CMOS technology in 2004 [24], as a
Fig. 3. Basic Component Layers of 3-D stacked CMVP

baseline core of our 3-D stacked CMVP. From [3], we can confirm that the SX-8 vector processor uses a large area for I/O interface logic; approximately 40% of the die area is used for SERializer/DESerialzer (Ser/Des) logic to keep high memory/pin bandwidth. The remaining part of the die area is spent for peripheral circuits and a vector core, which includes vector registers, vector pipe sets, a scalar unit, and an address control unit. The peripheral circuits and the vector core consume about 40% and 20% of the die area, respectively. Thus, it seems impossible to install many cores and a large shared memory on a chip, even though the technology scaling advances as steadily as ever. Hence, the 3-D stacked CMVP designed in this paper is composed of three kinds of layers; I/O layers, core layers, vector cache layers as shown in Figure 3.

The I/O layers contribute to increase off-chip memory bandwidth, and the core layers realize implementation of many cores on a die. The vector cache layers work for increasing the capacity of on-chip memory to compensate for the insufficient memory bandwidth of each core. The area of each layer is assumed as 21.7 x 18.05 mm², which is the same size with the SX-8 chip, and each layer is connected through TSVs with face-to-back stacking. By separating each function to the function-specified layers, the off-chip memory bandwidth, the number of cores and the capacity of vector cache are flexibly changed by stacking each layer.

A. I/O layer

On the I/O layer, we assume that only Ser/Des logic is implemented. First, we show the relationship between the area of Ser/Des logic and the off-chip memory bandwidth. In the case of SX-8, the off-chip memory bandwidth and the number of I/O pins are 64GB/sec and 8,210, respectively. The most latest vector processor SX-9 implemented in 65nm CMOS technology has a 256GB/sec off-chip memory bandwidth with 8,960 pins[3]. The number of transistors of SX-8 and SX-9 chips whose areas are almost the same are 88M and 350M, respectively. The area occupancies of Ser/Des logic in the processors are approximately 40% and 37%, respectively. Therefore we can roughly estimate that the off-chip memory bandwidth of a vector processor is proportional to the number of transistors of Ser/Des logic. Based on this estimation, we assume one Ser/Des layer has capability to realize 128GB/sec. That is, if we can use two Ser/Des layers for the 3-D stacked CMVP, it can realize 256GB/sec.

To keep a high sustained performance in execution of scientific and engineering applications, the vector processor requires that the ratio of the memory bandwidth to the arithmetic performance is higher than 4B/F[21]. Although the I/O layer can realize the 256GB/sec memory bandwidth, the arithmetic performance of the 3-D stacked CMVP can easily exceed 64Gflop/s, making it difficult to achieve 4B/F. Note that a single core of the latest vector processor SX-9 has already achieved 102.4Gflop/s. Accordingly, a mechanism to compensate for shortage of the B/F rate is necessary for a high sustained performance. Thus, the vector cache layer is adopted in the 3-D stacked CMVP.

B. Core Layer and Vector Cache Layer

The core layer includes two vector cores, and each vector core has four parallel vector pipe sets, each of which contains six types of vector arithmetic pipes (Mask, Logical, Add/Shift, Multiply, Divide, Square root), and 144 KB vector registers as shown in Figure 4. Based on an estimation using CACTI 5.3 [25], we assume that the vector cache layers consist of SRAM and DRAM with the 90nm technology. Each layer has 8MB and 32MB capacities, respectively. The vector cache layers are put between the I/O layers and the core layers.

The vector cache layer is designed based on plane design proposed in [26]. In this design, one sub-cache can fit in a layer as shown in Figure 5. Therefore, the size of each of sub-caches is varied from 256KB to 2MB as the total capacity of the vector cache is varied. Since each core is interconnected to the vector cache layers through 32 memory ports of a sub-cache by TSVs, at least 4096 TSVs (=64 bits x 32 ports) are required for connecting the vector layer and the vector registers on the core layer. Moreover, 4096 TSVs are also required for connecting the vector cache layer and an access control unit (ACU) on the core layer. As ACU generates eight load operations per cycle and the peak performance of the SX-8 vector processor is 16Gflop/s, the memory bandwidth between vector cache and each core reaches 4B/F.

V. PERFORMANCE EVALUATION

In this section, to clarify the effects of increasing the off-chips memory bandwidth and the number of cores by 3-D
integration technologies, we firstly evaluate the performance of CMVP without vector cache layers. Then, in term of sustained performance and energy consumption, CMVP with a vector cache is evaluate to confirm the potential of a 3-D stacked CMVP. Based on the performance evaluation, the tradeoff between performance and energy consumption is discussed to realize effective usage of a plenty of hardware resources given by 3-D integration technologies.

A. Evaluation Setup

An NEC SX trace-driven simulator that can simulate the behavior of the 3-D stacked CMVP architecture at the register-transfer level is used. The simulator is implemented based on the NEC SX vector architecture. The simulator accurately models a vector core of the SX-8 architecture; the vector unit, the scalar unit and the memory sub-system. The simulator takes a system parameter and a trace file of benchmark programs as inputs, and outputs instruction cycle counts. The specification of the 3-D stacked CMVP is shown in Table I. By introducing the 3-D Die stacking technology, the off-chip memory bandwidth is increased to 256GB/sec, the number of cores is possible at the maximum of 16, and the maximum capacity of the vector cache is 64MB. Since we assume the TSV has 2μm diameter with 30μm length [27], the access latency between cores and the vector cache is reduced to 70% of the cache access latency of conventional 2-D implementations.

The evaluation is carried out by using five leading applications, which are running at the SX-9 vector supercomputer systems at Tohoku University. Table II summarizes the characteristics of benchmark programs. Here, VOR indicates a vector operation ratio, and VL is the number of average vector elements per vector instruction, and PR is a parallel ratio of thread-level parallelism. More detail of these applications can be found in [28], [29], [30], [31], [32]. The benchmark programs are compiled by the NEC FORTRAN compiler, which can vectorize and parallelize the applications automatically. Then executable programs run on the SX trace generator to produce the trace files.

![Fig. 5. The Structure of Plane Designed Vector Cache Layers](image)

**TABLE I**

<table>
<thead>
<tr>
<th>Specification of 3-D Stacked CMVP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base System Architecture</strong></td>
</tr>
<tr>
<td><strong>Number of Cores</strong></td>
</tr>
<tr>
<td><strong>Number of Cores per Layer</strong></td>
</tr>
<tr>
<td><strong>Memory Size</strong></td>
</tr>
<tr>
<td><strong>Number of Banks</strong></td>
</tr>
<tr>
<td><strong>Vector Cache Implementation</strong></td>
</tr>
<tr>
<td><strong>Size of the Vector Cache</strong></td>
</tr>
<tr>
<td><strong>Size of the Sub-caches</strong></td>
</tr>
<tr>
<td><strong>Size of the Vector Cache per Layer</strong></td>
</tr>
<tr>
<td><strong>Cache Policy</strong></td>
</tr>
<tr>
<td><strong>Line Size</strong></td>
</tr>
<tr>
<td><strong>Cache Access Latency</strong></td>
</tr>
<tr>
<td><strong>Cache Bank Cycle</strong></td>
</tr>
<tr>
<td><strong>Off-chip Memory Bandwidth</strong></td>
</tr>
<tr>
<td><strong>Cache - Core Bandwidth</strong></td>
</tr>
</tbody>
</table>

**TABLE II**

<table>
<thead>
<tr>
<th>Benchmark Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Item</strong></td>
</tr>
<tr>
<td>PRF [28]</td>
</tr>
<tr>
<td>GPR [30]</td>
</tr>
<tr>
<td>APFA [31]</td>
</tr>
<tr>
<td>SFHT [29]</td>
</tr>
<tr>
<td>PBM [32]</td>
</tr>
</tbody>
</table>

B. Performance of the 3-D stacked CMVP without the Vector Cache

First, we evaluate the performance of the 3-D stacked CMVP with various number of cores by changing the number of core layers. As shown in the previous section, since one core layer includes two vector cores, 4 cores, 8 cores and 16 cores are implemented by two layers, four layers and eight layers, respectively. The off-chip memory bandwidth is also changed from 64GB/sec to 256GB/sec by stacking the I/O layers. In the case of using single I/O layer, the off-chip memory bandwidths are 64GB/sec and 128GB/sec. The double I/O layers realizes 256GB/sec as discussed in the previous section. The memory bandwidths per core are decreased as the number of cores increases, thus 4B/F rates per core is achieved in the cases of 2 cores with 128GB/sec of the off-chip memory bandwidth and 4 cores with 256GB/sec of the off-chip memory bandwidth.

Figure 6 shows the sustained performance of the 3-D stacked CMVP by using benchmark programs in Table II. We can confirm that the effects of introducing the I/O layers on the sustained performance. The performance 3-D stacked
CMVP increases as the off-chip memory bandwidth increases in all the cases. The performance improvement by the off-chip memory bandwidth improvement is summarized in Table III. The improvement rates are evaluated based on each 64GB/sec case. We can also confirm that the performances of all the cases except PBM in Figure 6 (c) scale well as the number of cores increases. The reason is that the memory bandwidths per core of 64GB/sec and 128GB/sec cases are relatively low compared to the 256GB/sec case. For example, the B/F rates per core of 16 cores with 64GB/sec case are 0.25, whereas those of 16 cores of 256GB/sec reach 2B/F.

In the PBM case in Figure 6 (c), the performances of the 3-D stacked CMVP are dropped as the number of core increases. This is because redundant memory accesses become remarkable as the number of cores increases, since all cores on the 3-D CMVP access the same data on the main memory. In addition, as shown in Figures 6 (a) and (b), the performance of PBM does not scale well as the number of core increases. The penalty of redundant memory accesses becomes larger than the benefit of increasing the number of cores. Therefore, the sustained performance of the PBM is degraded as the number of cores increases. In the next section, we discuss the effects of the vector cache to remove redundant off-chip memory accesses.

C. Effects of the Vector Cache on Sustained Performance

Figure 7 shows the effects of 64MB vector cache MSHR mechanisms on the sustained performance, which is composed of 8 vector layers. The relative performances are normalized by corresponding configurations in Figure 6. In this evaluation, two cores configuration with a memory bandwidth of 256GB/sec are not considered because these configurations have an enough memory bandwidth per core of 8B/F. In the 64GB/sec case, the vector cache effectively works compared to other two cases. Since the off-chip memory bandwidth per core is quite low in this case, the vector cache contributes to an improvement in the sustained memory bandwidth. In the case of 128GB/sec, though the effectiveness is lower than the 64GB/sec case, the performance still increases by the vector cache. However, in the case of 256GB/sec, the effects of the vector cache cannot be observed except the PBM cases. As mentioned in the previous section, PBM generates redundant memory accesses proportional to the number of cores. The vector cache includes the MSHR mechanisms effectively eliminates redundant memory accesses by caching the data, which are accessed by all cores. Therefore the performance of PBM mainly improved by the MSHR mechanism.

D. Discussions

Even though, the recent 3-D integration technology has a potential to aggressively stack many layers, the less number of layers are more preferable not only for achieving high yields but also reducing implementation costs and power consumptions. Thus, we confirm the minimum required cache capacity to keep a certain performance of the 3-D stacked CMVP. Figure 8 shows a relative performance of APFA by changing a the size of vector cache and the number of I/O layers. The relative performance is obtained by normalizing the sustained performance of a CMVP by that of a single core processor which has a 64GB/sec off-chip memory bandwidth without the vector cache, as same as the SX-8 processor. From this fact, we can confirm that 8MB vector cache has an enough potential to keep the high performance. Thus, to keep the highest sustained performance, 16 cores, an 8MB vector cache with the 128GB/sec and 256GB/sec off-chip memory bandwidths (eight core layers, one vector cache layer and two I/O layers) seems the best configuration for this application.

However, the performance of the two cases is almost the same (less than 1% difference). Figure 9 shows the energy consumption of APFA with an 8MB vector cache. The energy consumption the 3-D stacked CMVP takes into account those of the core layers, the vector cache layer and the I/O layers. In Figure 9, we can confirm that the energy consumption of 16 cores with the 128GB/sec off-chip memory bandwidth is 14% less than that of 16 cores with the 256GB/sec off-chip memory bandwidth. In addition, in the case of 16 cores with the 128GB/sec off-chip memory bandwidth, we can also confirm that the 8MB vector cache achieves a 51% performance improvement. In short, to achieve energy efficient performance improvements, introducing the vector cache is more effective than increasing the off-chip memory bandwidth. This fact can also be confirmed by 4 cores and 8 cores cases. Furthermore, since the vector cache significantly reduces the off-chip memory accesses, the performance improvement and energy reduction are shown in all the cases in Figure 9.
TABLE III

<table>
<thead>
<tr>
<th></th>
<th>2 cores</th>
<th>4 cores</th>
<th>8 cores</th>
<th>16 cores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8B/F</td>
<td>16B/F</td>
<td>8B/F</td>
<td>16B/F</td>
</tr>
<tr>
<td>PRF</td>
<td>28%</td>
<td>151%</td>
<td>63%</td>
<td>128%</td>
</tr>
<tr>
<td>APFA</td>
<td>4%</td>
<td>76%</td>
<td>29%</td>
<td>33%</td>
</tr>
<tr>
<td>SFHT</td>
<td>72%</td>
<td>174%</td>
<td>93%</td>
<td>188%</td>
</tr>
<tr>
<td>GPR</td>
<td>78%</td>
<td>209%</td>
<td>99%</td>
<td>233%</td>
</tr>
<tr>
<td>PBM</td>
<td>78%</td>
<td>267%</td>
<td>104%</td>
<td>195%</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

Aiming at realizing efficient computing and clarifying the design space of the 3-D integration technology, the 3-D stacked CMVP is designed and evaluated. Due to the area and off-chip memory bandwidth limits, the 3-D integration technology has been introduced to implement the 3-D stacked CMVP. The 3-D stacked CMVP consists of I/O layers, core layers and vector cache layers. This paper also confirms the effects of large vector cache, which realized by the 3-D integration technology by performance evaluation using real scientific and engineering applications. The vector cache layer can dramatically increase the capacity of the vector cache, which can effectively store reusable data, to compensate for a decreasing B/F rate per core. Furthermore, the vector cache layer reduces the energy consumption of 3-D stacked CMVP by declining the number of off-chip memory accesses. Since the I/O layer improves the off-chip memory bandwidth, the performance of 3-D stacked CMVP is also improved.

Based on the performance evaluation with practical scientific and engineering applications, the effective use of a vast amount of hardware resource in terms of performance and energy consumptions is discussed. A 16 cores 3-D stacked CMVP with an 8MB vector cache realizes the most energy efficient computation. The experimental results indicate that it achieves a 52% performance improvement with a 14% energy reduction compared to that without the vector cache. The results of performance evaluation have clarified that introducing the vector cache is more effective than increasing the off-chip memory bandwidth to achieve energy efficient performance improvements.

In this paper, we just consider one configuration of the vector cache. However, to realize more powerful computing environments with extremely high memory bandwidth, a novel 3-D stacked vector cache configuration should be considered. Besides, we will carry out the performance evaluation with more large data set, to confirm the requirement of the future HPC applications in an era of exascale computing[33][34].
ACKNOWLEDGMENT

The author would like to thank Professor Mitsumasa Koyanagi of Tohoku University and Jun Inasaka and Dr. Shintaro Momose of NEC cooperation for valuable discussion on this research. This research was partially supported by Grant-in-Aid for Scientific Research (Grant-in-Aid for Young Scientists (B) No.22700044) and (Grant-in-Aid for Scientific Research (B) No. 22300013), the Ministry of Education, Culture, Sports, Science and Technology. This research was also partially supported by Core Research for Evolutional Science and Technology (CREST), Japan Science and Technology Agency (JST). This work was also partially supported by Grant-in-Aid for JSPS Fellows, Japan Society for the Promotion of Science, No. 21007908.

REFERENCES


