ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.10, No.4, 1–6

Versatile stream buffer architecture to exploit the high memory bandwidth of 3-D IC technology

Hong-Yeol Lim[a)] **and Gi-Ho Park**[b)]

Department of Computer Engineering, Sejong University,
98, Gunja-Dong, Gwangjin-Gu, Seoul, 143–747, Republic of Korea

a) *hylim@sju.ac.kr*

b) *ghpark@sejong.ac.kr, correspoding author*

Abstract: Three-dimensional (3-D) integration technology provides various architectural opportunities including huge memory bandwidth. This paper proposes versatile stream buffer architecture to work as a secondary victim cache as well as the conventional stream buffer. The versatile stream buffer utilizes empty spaces to exploit massive memory bandwidth provided by 3-D integration technology and to reduce memory access frequency. Performance evaluation results show that the proposed mechanism with a 16 KB stream buffer and a 4 KB victim cache can achieve better performance than the conventional L2 cache with the capacity of 256 KB and 2 MB by 10% and 3%, respectively. The proposed mechanism reduces the miss rate by about 12% more than the conventional L2 cache with the capacity of 256 KB.

Keywords: 3-D integration technology, stream buffer, victim cache

Classification: Integrated circuits

References

[1] N. P. Jouppi, "Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffer," *ISCA-17*, pp. 364–373, May 1990.

[2] S. Palacharla and R. E. Kessler, "Evaluating Stream Buffer as a Secondary Cache Replacement," *ISCA-21*, pp. 24–33, April 1994.

[3] K. Inoue, K. Kai, and K. Murakami, "Dynamically Variable Line-Size Cache Exploiting High on-chip Memory Bandwidth of Merged DRAM/Logic LSIs," *HPCA-5*, pp. 218–222, Jan. 1999.

[4] T. Ono, K. Inoue, and K. Murakami, "Adaptive Cache-Line Size Management on 3D Integrated Microprocessors," *ISOCC 2009*, pp. 472–475, Nov. 2009.

[5] D. H. Woo, N. H. Seong, D. L. Lewis, and H. H. S. Lee, "An Optimized 3-D Stacked Memory Architecture by Exploiting Excessive, High-Density TSV Bandwidth," *HPCA-15*, pp. 1–12, Jan. 2010.

[6] C. Liu, I. Ganusov, M. Burtscher, and T. Sandlip, "Bridging the Processor-Memory Performance Gap with 3D IC Technology," *Design & Test Computers*, pp. 556–564, Dec. 2005.

[7] J. Sharkey, "M-Sim: A Flexible, Multithreaded Architectural Simula-

DOI: 10.1587/elex.10.20120971
Received December 28, 2012
Accepted January 17, 2013
Published February 27, 2013

tion Environment," Technical Report CS-TR-05-DP01, Department of Computer Science, State University of New York at Binghamton, 2005.
[8] A. Jeleel, "Memory Characterization of Workloads Using Instrumentation-Driven Simulation-A Pin-Based Memory Characterization of the SPEC CPU 2000 SPEC 2006 Benchmark Suites," Intel VSSAD Technical Report, 2007.
[9] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," *Workload Characterization*, pp. 3–14, Dec. 2001.
[10] S. Thoziyoor, "CACTI 5.3," HP Laboratories, Palo Alto, CA, 2008.

1 Introduction

The semiconductor and computer system areas are investigating three dimensional (3-D) integration technology as an approach to meet tight performance and power constraint. This technology provides the opportunity for huge memory bandwidth and flexible placement of memory components to devise new system architectures for exploiting these characteristics. One of mechanisms to exploit this advantage is a data prefetching technique. This is one of the well-known key techniques to improve the performance of the memory system effectively.

This paper proposes a versatile stream buffer architecture to work as a secondary victim cache as well as a conventional stream buffer. The proposed versatile stream buffer architecture with a victim cache utilizes the empty space of the stream buffer and stores the evicted data from the victim cache to exploit high memory bandwidth and to reduce memory traffic.

The performance evaluation results show that the proposed 16 KB stream buffer with a 4 KB victim cache can achieve about 10% and 3% better performance than the conventional 256 KB L2 cache and 2 MB L2 cache. The proposed versatile stream buffer with victim cache reduces the miss rate by about 12% more than conventional 256 KB L2 cache on average.

This paper is organized as follows. Section 2 explains related work. The proposed versatile stream buffer architecture with victim cache is described in Section 3. Section 4 presents the simulation environment and performance evaluation result. Section 5 concludes the paper.

2 Related work

Various studies have been proposed to efficiently utilize memory bandwidth of computer system. The stream buffer structure and victim cache as a small buffer for the L1 direct-mapped cache are introduced in [1]. In [2], Palacharla et al. adopted the concept of the original stream buffer into the L2 cache in order to reduce long off-chip memory access latency.

Adaptive prefetching mechanisms have been proposed as well. Inoue et al. proposed a mechanism that adjusts fetch size based on the access pattern of the cache block in [3]. Recently, various schemes have been investigated to

DOI: 10.1587/elex.10.20120971
Received December 28, 2012
Accepted January 17, 2013
Published February 27, 2013

exploit plentiful memory bandwidth advantage provided by 3-D integration technology. Ono et al. [4] proposed a software-controllable mechanism to adjust fetch sizes dynamically based on the profiling information gathered during compiling time for the 3-D integration technology. In [5], Woo et al. proposed a SMART-3D architecture, which efficiently utilized the TSVs to reduce long latency of fetching and write-back in an L2 cache. Liu et al. [6] investigated various conventional schemes including a stream buffer to bridge the processor and memory performance gap based on the 3-D ICs. This paper introduces versatile stream buffer architecture with a victim cache to exploit massive memory bandwidth and to reduce memory traffic.

3 Versatile stream buffer architecture with victim cache

Fig. 1 presents the proposed system architecture of the versatile stream buffer with a victim cache. The stream buffer and the victim cache are located below the L1 direct-mapped cache and constructed with a small fully-associativity cache. The stream buffer has 16-way buffers with the 16-entries of 64 bytes block. The victim cache has one-way with the 64-entries of 64 bytes block. The stream buffer entries have a valid bit. This bit indicates whether the buffer entry contains valid data or not. It is used to find the buffer entry to store the evicted data from the victim cache.

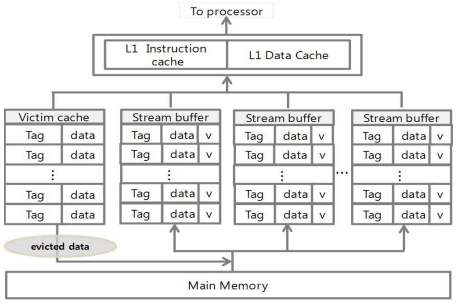

Fig. 1. Versatile stream buffer with a victim cache.

The versatile stream buffer and the victim cache store the prefetched blocks and victim blocks from the L1 cache as does the usual stream buffer and victim cache as proposed in [1] and [2], respectively. The stream buffer is constructed bigger than the conventional stream buffer to prefetch more blocks to exploit the high memory bandwidth of 3-D integration technology. Even though this aggressive prefetching usually provides better performance and incorrect prefetching is interrupted by other cache misses.

Fig. 2 shows the number of invalidated blocks when the stream buffer capacity is 16 KB. As shown in Fig. 2, the invalidated blocks are about 70% of total 256 entries on average. These invalidated blocks are remains useless empty spaces until new prefetch blocks are inserted. To utilize this empty space, the versatile stream buffer stores the evicted data from the victim cache into the invalidated entries of the stream buffer.

DOI: 10.1587/elex.10.20120971
Received December 28, 2012
Accepted January 17, 2013
Published February 27, 2013

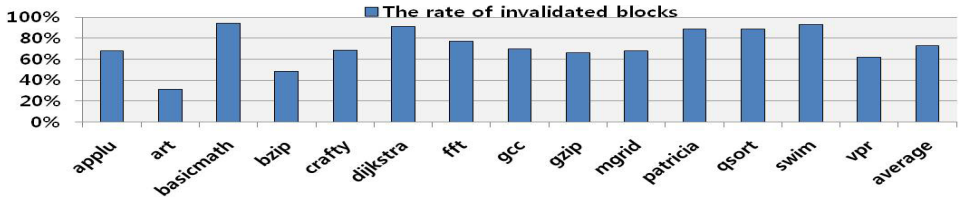

Fig. 2. The rate of invalidated blocks in the stream buffer.

4 Operation model of proposed architecture

The operation model of the proposed versatile stream buffer architecture is shown in Algorithm 1.

Algorithm 1: Operation model of proposed versatile stream buffer architecture

```
1.  If L1 cache miss then
      {
    If victim cache hit then
      {
        Transfer the hit data to L1 cache and processor (hit
    data from victim cache)

        If conflict in the L1 cache
          Cache line and matching victim cache line are s
    wapped
      }

2.  Else  //If victim cache miss then
      {
     If stream buffer miss then
       {
          Transfer miss data to L1 cache and processor
    from main memory
          Select the stream buffer way by LRU policy
          Prefetching successive data to selected stream
    buffer way
       }
    Else  //If stream buffer hit then
          Transfer data to L1 cache and processor (hit
    data from stream buffer)

    If conflict in the L1 cache then
      {
          Transfer data to victim cache (evicted data
    from L1 cache)
          If conflict in the victim cache then
          {
            Select the invalidation entry in stream
    buffer by MRU policy
            Insert evicted data from victim cache
          }
          Else //If not conflict in victim cache then
            Insert into victim cache (evicted data from
    L1 cache)
      }
    }
```

When an L1 cache miss occurs, the stream buffer and victim cache are searched. If the access is hit in the stream buffer or victim cache, the cache block is transferred to the L1 cache and the processor from the stream buffer or victim cache.

In the case of a miss in the stream buffer and victim cache, a stream buffer way to be replaced is selected based on the LRU, and the prefetching is performed for the data after the cache block in which the cache miss occurs. If the access is a hit in the victim cache then the cache block is swapped between the L1 cache and victim cache. In addition, the proposed mechanism stores the replaced entry from the victim cache into the empty stream buffer entry selected by the Most Recently Used (MRU) among the invalidated entries. This insertion policy guarantees that the MRU entry can be stored as long as possible.

5 Simulation environments and performance evaluation

Performance evaluation was performed based on M-Sim simulator [7] with SPEC CPU 2000 [8] and Mibench [9] benchmarks. The system parameter for simulation is presented Table I. The access latency of the L1 and L2 cache memories, the stream buffer and the victim cache are obtained from

DOI: 10.1587/elex.10.20120971
Received December 28, 2012
Accepted January 17, 2013
Published February 27, 2013

Table I. System Configuration.

L1 Cache (Instruction/Data Cache)	Capacity : 16 KB/16 KB (I/D), Block size : 64 B, Associativity : 4-way, Hit latency : 2 cycle
Main Memory	Latency (first latency - next chunk latency) : 54 - 2, Memory bandwidth : 512 bit
Stream Buffer and Victim Cache	Capacity : 256 B˜ 8 MB, Hit latency : 1˜ 258 cycle, Block Size : 64 B

the CACTI 5.3 tool [10].

Fig. 3 shows the performance of victim caches and stream buffers having various capacities. As shown in Fig. 3, the best performance of a victim cache and a stream buffer can be obtained when the capacity is 128 KB for both cases. Even though the stream buffer and victim cache having capacity with 128 KB can achieve the best performance, we select the 4 KB for a victim cache and 16 KB for a stream buffer. It is mainly because the difference of performance with 128 KB capacity is negligible and the access latency of fully-associative 128 KB buffer is too large. We have analyzed the performance of the proposed architecture according to the number of ways and entries of the stream buffer. The performance differences among the 8-way 32-entry, 16-way 16-entry and 32-way 8-entry stream buffers are only about 3–4% in CPI. The 16-way 16-entry stream buffer delivers the best performance among these configurations. The configuration of the stream buffer is determined as a 16-way 16-entry buffer based on this analysis. The victim buffer is constructed as a one-way fully associative buffer as proposed in [1].

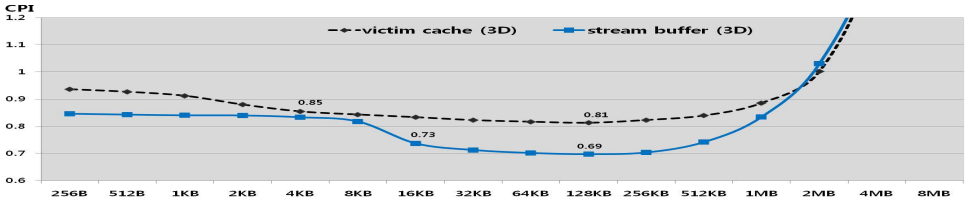

Fig. 3. Performance of the victim cache and the stream buffer.

The performance evaluation results of the proposed versatile stream buffer are presented in Fig. 4. It shows that the proposed versatile stream buffer having a 16 Kbytes capacity with a 4 Kbytes victim cache can achieve per-

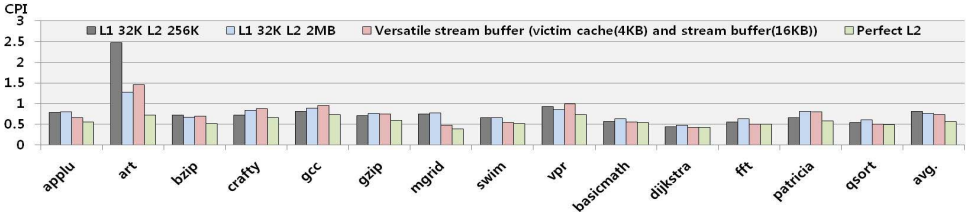

Fig. 4. Performance of versatile stream buffer architecture.

DOI: 10.1587/elex.10.20120971
Received December 28, 2012
Accepted January 17, 2013
Published February 27, 2013

formance improvement over the conventional L2 cache with the capacity of 256 Kbytes and 2 Mbytes by about 10% and 3% respectively.

Fig. 5 shows the performance of the proposed versatile stream buffer architecture with victim cache and conventional stream buffer with victim cache. The proposed architecture having 16 KB stream buffer and 4 KB victim cache can achieve a performance improvement of about 6% over the conventional stream buffer and victim cache with the same capacity. The proposed architecture can achieve a performance improvement of about 3% more than the conventional stream buffer and victim cache with twice the capacity, i.e., 32 Kbytes and 8 Kbytes each as well.

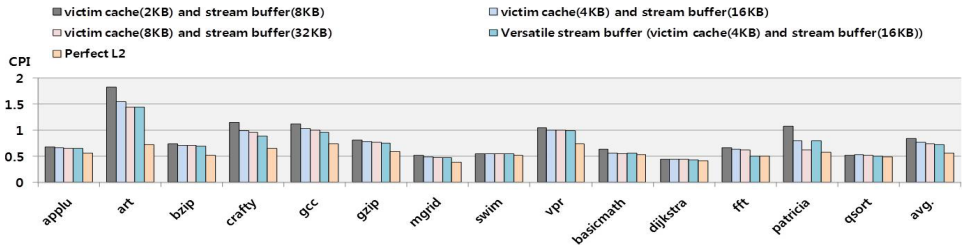

Fig. 5. Performance of versatile stream buffer architecture.

6 Conclusion

This paper proposed a versatile stream buffer with a victim cache to exploit the enormous memory bandwidth provided by 3-D integration technology and to reduce memory traffic. The proposed versatile stream buffer architecture can be adopted into a conventional L2 cache structure as well.

The proposed mechanism can achieve performance improvement over the conventional L2 cache with the capacity of 256 KB and 2 MB by about 10% and 3%, respectively. The proposed architecture reduced the miss rate more than the conventional 256 KB L2 cache by about 12% on average.

The result provides that a more sophisticated mechanism to exploit enormous memory bandwidth and to reduce memory loss and effective memory traffic is required rather than just extending the conventional mechanism likes increasing cache block size. One of most important future research will be the analysis and adaptation of the versatile stream buffer into the multi-core system architecture.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No.2010-0006785). A preliminary version of this paper was published in Advanced Information Technology and Sensor Application (AITS) 2012.

DOI: 10.1587/elex.10.20120971
Received December 28, 2012
Accepted January 17, 2013
Published February 27, 2013