Dynamic Segmented Bus for Energy-Efficient Last-Level Cache in Advanced Interconnect-Dominant Nodes

Mahta Mayahinia, Tommaso Marinelli, Zhenlin Pei, *Graduate Student Member, IEEE*, Hsiao-Hsuan Liu, Chenyun Pan, *Senior Member, IEEE*, Zsolt Tokei, Francky Catthoor, *Fellow, IEEE*, and Mehdi B. Tahoori, *Fellow, IEEE*

***Abstract*—To deal with stagnated performance and energy improved by successive technology scaling, system-technology co-optimization (STCO) comes as a rescue which involves the co-optimization of the important system parameters from the high-level application all the way down to the low-level technology. This article addresses the interconnect dominance issue in advanced nodes as a bottleneck in energy-efficient static RAM (SRAM)-based last-level cache (LLC) and aims to mitigate it through an STCO mechanism. Our main approach in this work is the utilization of a workload-aware controlled dynamic segmented bus (DSB) as the intramacro (interbanks) interconnect. Based on our results, our approach can improve the energy efficiency of the SRAM-based LLC by an average of 35%.**

***Index Terms*—Advance node, energy efficiency, interconnect dominance, last level cache, segmented bus, SRAM, system-technology co-optimization.**

I. Introduction

FEATURE size reduction of the Front End of the Line (FEoL) and Back End of the Line (BEoL) is no longer sufficient for high-performance energy-efficient VLSI designs in sub-10-nm advanced technology nodes. Since the resistive-capacitive (RC) delay of the interconnect increases leading to a dominant effect on the overall system energy and latency [1]. System-technology co-optimization (STCO) is a complementary solution to device scaling to achieve higher performance and energy efficiency in which the critical parameters, from the high-level applications all the way down to the low-level technology parameters are co-optimized together.

Manuscript received 12 August 2024; accepted 12 August 2024. This manuscript was recommended for publication by A. Shrivastava. *(Corresponding author: Mahta Mayahinia.)*

Mahta Mayahinia and Mehdi B. Tahoori are with the Departement of Computer Science, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany (e-mail: mahta.mayahinia@kit.edu).

Tommaso Marinelli is with the Department of Computer Architecture and Automatic Control, Universidad Complutense de Madrid, 28040 Madrid, Spain, and also with Katholieke Universiteit Leuven, 3000 Leuven, Belgium.

Zhenlin Pei is with the Department of Electrical Engineering, The University of Texas at Arlington, Arlington, TX 76010 USA.

Hsiao-Hsuan Liu and Francky Catthoor are with the Department of Electrical Engineering, Katholieke Universiteit Leuven, 3000 Leuven, Belgium, and also with IMEC v.z.w., 3001 Leuven, Belgium.

Chenyun Pan is with the Department of Electrical Engineering, The University of Texas at Arlington, TX 76010 USA.

Zsolt Tokei is with the Silicon Technology System Unit, IMEC v.z.w., 3001 Leuven, Belgium.

Digital Object Identifier 10.1109/LES.2024.3444711

In this article, we focus on improving the energy efficiency of the static RAM (SRAM)-based cache memory. In the modern processors, due to the high-integration density of the advanced nodes, the capacity of SRAM-based cache memory can be unprecedentedly large, significantly affecting the overall energy and latency. Therefore, addressing the large RC delay of the interconnect is a crucial step in the design of high-performance and energy-efficient large cache memories.

The impact of the different levels of the interconnects on the SRAM functionality is not similar. Subarray-level interconnects word-line (WL) and bit-line (BL) fabricated on the lower-metal layer affect the noise margin [2], [3], [4], [5]. Since intramacro (interbank) interconnects are fabricated on higher-metal layers (M5-M8), due to the higher-RC delay in the advanced technology nodes, such intermediate interconnects dominate the total latency and energy. Nevertheless, this issue has not been addressed in the existing literature.

To fill the gap, intramacro, henceforth, interbank interconnects of a last-level cache (LLC) are the main focus of this work. First, to reduce the high-capacitive load of the long wires fabricated on higher-metal layers and consequently energy reduction, we aim to disable the unused segments of the interbank interconnect, using dynamic segmented bus (DSB) structure. Also, to achieve further energy saving, this hardware solution needs to be complemented with a run-time strategy to control the activation of the segments in a workload-aware fashion. Besides DSB and its control scheme, we also aim to map the frequently used data into the banks with shorter segments. This way, the high-rate data exchange can be concentrated on short segments.

II. Segmented Bus in Victim Last-Level Cache

A victim cache accommodates the evictions of the faster-level cache. Therefore, in the case of a 'miss' event at the faster-level cache, the block can be provided through the victim cache and not the main memory. Fig. 1 shows the interaction of the LLC with the CPU, faster cache levels, and the main memory.

The impact of the technology node in SRAM-based cache memory is critical, mainly due to the rapid increase of the RC delay of the interconnect with technology scaling as indicated by Table I. Due to the high-RC delay of the interbank interconnects, compared to the farther banks, the access latency and energy of the closest banks are considerably less.

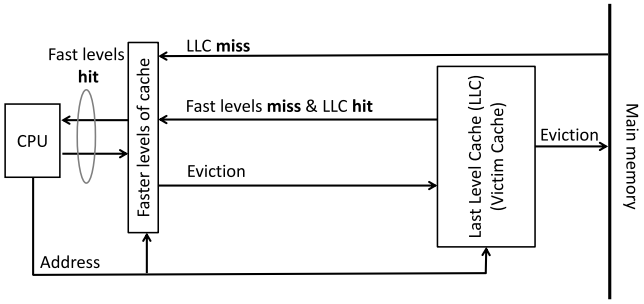


Fig. 1. Last-level cache (LLC) (as a victim cache) interactions with the CPU, faster cache levels, and the main memory.

TABLE I
COMPARISON OF THE ENERGY AND LATENCY OF 128-MB SRAM-BASED CACHE, ORGANIZED IN 16 BANKS, AND DATA WIDTH OF 512 BITS

CMOS Interconnect	RC delay (Interconnect)	Optimum Latency [ns/mm]	Optimum Energy [pJ/mm]	Cycle Time [ns]	Farthest bank		Closest bank	
					Latency	Energy	Latency	Energy
- 90 nm - 90 nm CD	~20 ps/mm	0.14	392	15.56	5.39 ns 1 Cycle	59.3 nJ	43.8 ps 1 Cycle	44.1 nJ
- 40 nm - 40 nm CD	~80 ps/mm	0.17	206	10.72	2.97 ns 1 Cycle	14.1 nJ	122 ps 1 Cycle	10.7 nJ
- 32 nm - 32 nm CD	~200 ps/mm	0.19	179	9.39	2.68 ns 1 Cycle	9.82 nJ	172 ps 1 Cycle	7.94 nJ
- 22 nm - 22 nm CD	~500 ps/mm	0.22	143	6.29	2.07 ns 1 Cycle	5.32 nJ	218 ps 1 Cycle	4.14 nJ
- 3 nm - 16 nm CD	~80 ns/mm	1.11	88	0.33	6.74 ns 21 Cycle	0.57 nJ	2.26 ns 7 Cycles	0.22 nJ

To make a connection between technology and architecture, we compare a 128 MB, side-pin assess, 16-bank cache in different technology nodes. Fig. 2 shows such a structure and the distances from the root access pin corresponded to the 16-nm critical wire dimension (CD) as used in 3-nm CMOS technology. Table I also shows a comparison between different technologies using the CACTI [6] framework. As shown in Table I, in the case of 16-nm CD interconnect technology, the group of banks that are located closest to the root access pin, can be accessed 14 cycles faster and requires less than half the energy compared to the farthest bank.

Besides, in a memory macro, multiple banks need to be connected to the root access pin. For such connectivity, the E-tree structure is a viable option primarily due to its feature in equalizing the propagation delay across the banks [7]. An E-tree structure is also shown in Fig. 2. However, as shown in Table I, equalizing the access latency within the banks forces the worst-case latency on all the banks, extremely degrading the performance. Exploiting the inherent asymmetry in access latency and energy in the case of advanced technology nodes is our aim in this article. For this, we leverage the segmented bus structure—an established VLSI interconnect [8], [9], [10], [11]—as an interbank interconnection in a large SRAM-base LLC.

The basic idea of the segmented bus is the division of the entire bus into smaller segments. This way, the active capacitive load of the long wire decreases, contributing to a lower-switching power and subsequently, lower-energy consumption [8]. Besides the segmented bus, the works presented in [9] and [10] also addressed the significance of physical floorplanning on the energy reduction of a communication architecture. Further, these works ([9], [10]) introduced the concept of DSB, in which the various segments are not necessarily equal to each other. Moreover, a software-controlled fashion for the DSB was also proposed in [12]. More recently, the work presented in [11] utilized the DSB for global synapse communication in a neuromorphic fabric.

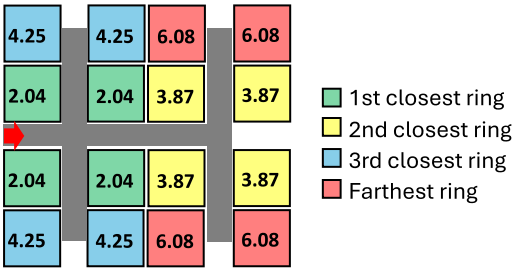


Fig. 2. 16-bank SRAM-based memory macro fabricated in 16-nm CD, the numbers inside the banks (in mm) show the distance between the bank and the root access pin, the red arrow indicates the root access pin location, and the gray line shows the E-Tree [7].

The vast amount of literature utilizing the DSB makes this structure a well-established energy-efficient yet flexible interconnection network. Therefore, we also aim to utilize it as a substitute for the E-tree mainly in large cache memories to improve the overall energy efficiency.

III. IMPLEMENTATION OF THE SEGMENTED BUS IN VICTIM LAST-LEVEL CACHE

To address the high-energy consumption of the interconnect network in a large memory macro fabricated in advanced technology nodes, we have two objectives as follows.

1) Fully utilize the bank's capacity by condensing the accesses of a workload into a minimum number of banks and then map them to the banks with the shortest distance to the root access pin. This ensures the minimization of energy and latency of access to such data.
2) Disabling the unused (farther banks) in a workload-aware fashion with the help of DSB. Therefore, only the minimum subset carrying application data is activated per each access, not the entire interconnection network, further reducing energy.

To achieve these objectives, we first obtain the total number of LLC accessed addresses as the key characterization of the workload and use it to cluster the workloads. Then, we use the output of this clustering for floorplanning and programming of the switches required in the structure of the DSB.

A. Workload Characterization

The input for the workload characterization is the LLC trace per each workload and its output is the number of accesses per each address. To obtain the LLC trace, we use gem5, an event-driven micro-architectural simulator. Then, by performing the statistical analysis on the workload traces, the total number of accessed addresses and their frequency can be obtained.

Though the essence of our proposed methodology can be generalized to any number of banks, we carry out our investigation on a 16-bank LLC macro. So, we cluster the workloads into 16 classes as follows: Class 1 consists of workloads whose total accessed addresses can be accommodated in 1 LLC bank, all the way to Class 16 which includes the workloads that need all the 16 banks of LLC to accommodate their accesses. Condensing the workload accesses to the minimum number of banks with the shortest distance from the root access pin can be performed at the design-time compiler stage.

B. DSB Floorplanning

With the discussed clustering (Section III-A), we determine the bank size as the granularity for floorplanning and

controlling of the DSB. To fulfill both objectives, we assume there is a communication path from the root access pin to the bank N if and only if bank $N-1$ too has a communication path to the root access pin.

From a recursive perspective, establishing communication between bank $N-1$ and root access pin is the prerequisite of the communication path between bank N and root access pin and so on. This statement implies a hierarchical structure to the floorplan which can be defined as follows. A memory macro consists of multiple equidistant bank groups, we refer to them as *rings* (see Fig. 2). At the next level of the hierarchy, we have *subrings*. So, each ring consists of multiple subrings, and finally, subrings consist of multiple banks. Such a hierarchical floorplanning is shown in Fig. 3(a). Please note that each of the switches is bidirectional, accounting for both the read and write accesses.

Besides, in the case of LLC involvement in servicing any access, the first step is mapping of a corresponding address in the LLC hierarchy. In the case of having 16 banks, 4 bits of the address can be used to determine the bank in a conventional interconnect structure. In a DSB utilization, however, the class of the workload, i.e., the number of its required banks is already known due to the workload clustering as discussed in Section III-A. Therefore, to perform such address mapping, the only requirement is the substitution of the workload class for the 4-bit bank representative, which indicates a negligible change in the routing scheme.

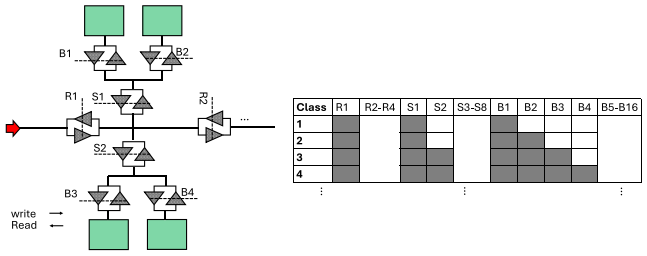
To realize a low-overhead run-time-applicable scheme for controlling the DSB as an interbank interconnection, we can have a lookup table as shown in Fig. 3(b). Please note that the storage overhead of the programming lookup table shown in Fig. 3(b) is only 28 bits per workload class which accumulated to the total value of 56 B. This small storage overhead guarantees a negligible energy overhead as well.

The initial recursive assumption and its subsequent hierarchy bring the scalability to the DSB. This way, extending the DSB by adding any ring, subring, or bank can be done by simply instantiating the core structure either for the switches or for the controlling lookup table as shown in Fig. 3. Due to this hierarchical structure, a bank cannot be enabled unless its corresponding subring and ring are also enabled.

IV. Results and Discussion

First, we show the results of the electrical-level simulation using SPICE for obtaining the energy overhead of the DSB. Further, to evaluate the effectiveness of the DSB structure, the LLC access traces are required. To obtain them, we execute the most representative interval of 100 million instructions of eight realistic workloads from the SPEC 2017 benchmark suite with the parameters outlined in Table II and using gem5. To make the experiments and analysis computationally feasible, we select these eight workloads as a representative subset of the SPEC2017 benchmark suite. These selected workloads do not behave similarly [13] and have different degrees of computing and memory intensity natures [14]. To show the effectiveness of the DSB on the overall energy reduction, we obtain the per-workload energy consumption and reason regarding the achieved energy gains.

1) Circuit-Level Analysis: To obtain and optimize energy overhead induced by the DSB switching ($E_{\text{DSB}}^{\text{Overhead}}$), we perform the electrical-level simulation on a transmission gate (TG)-based switch which loads a specific length of the wire



(a) (b)

Fig. 3. (a) Multibank last-level cache (LLC) and the proposed positions for the switches in a dynamic segmented bus (DSB) structure, R, S, and B stand for ring, subring, and bank, respectively, and (b) controlling lookup table the DSB switches per application class.

TABLE II
Simulation Assumptions and Parameters

CPU characteristics					
Model				X86 Out-of-Order CPU	
Clock Frequency				3 GHz	
Physical address				64 bits	
Cache hierarchy characteristics					
Cache line size				64 B	
Level	Tech.	Size	Assoc.	Access latency (Number of cycles)	Number of banks
L1I/D	SRAM	32 kB	8	3	–
L2	SRAM	512 kB	16	9	4
L3	SRAM	128 MB	16	33	16
CMOS technology				3 nm	
Interconnect characteristics					
Wire (resistance, capacitance) per unit length (with repeater insertion consideration)				(4.34 Ω/mm, 192 fF/mm)	
Optimum energy per unit length per bit				171.8 fJ/mm	
Via resistance (R_{via})[13], [14]				57.6 Ω	

segment. For instance, in Fig. 3(a), the R1, R2, R3, and R4 is loading the segment lengths of 2.04, 1.83, 0.38, and 1.83 mm, respectively. To optimize $E_{\text{DSB}}^{\text{Overhead}}$, we increase the size of the TG-based switches until reaching an optimum energy value. Fig. 4 shows such a procedure to optimize $E_{\text{DSB}}^{\text{Overhead}}$ which is linearly proportional to its corresponding segment length.

In this work, the considered SRAM-based LLC has utilized three metal layers (M1-M3) in its subarray levels and two metal layers (M4, M5) for its intrabank interconnects. As shown in Fig. 3(a), the proposed DSB also has two horizontal and vertical interconnects, which require two metal layers (M6, M7). Table II shows the corresponding resistance and capacitance (R, C) for M6 and M7. In our electrical-level simulation, we also consider the effect of the via (four via connections have been considered) [15], [16]. To investigate the effect of the via, we sweep its resistance (R_{via}) from 0 to its twice nominal value (refer to Table II), and as clearly shown by Fig. 4, the variation of the R_{via} has a negligible impact on the optimized energy overhead of the DSB.

2) Workload-Level Analysis: Table III shows the summary of these workloads with respect to the total number of unique accessed addresses and consequently, the number of occupied banks in LLC. As shown in Table III, except workload 649, all the other evaluated workloads tend to occupy the first, i.e., the innermost ring. Therefore, using the DSB can reduce energy consumption by turning off the unused segments and mapping all the addresses to the innermost ring.

3) Calculating Perworkload Energy Consumption: Finally, by having both the traces from the workload-level analysis, the optimized energy consumption of the interconnect, and

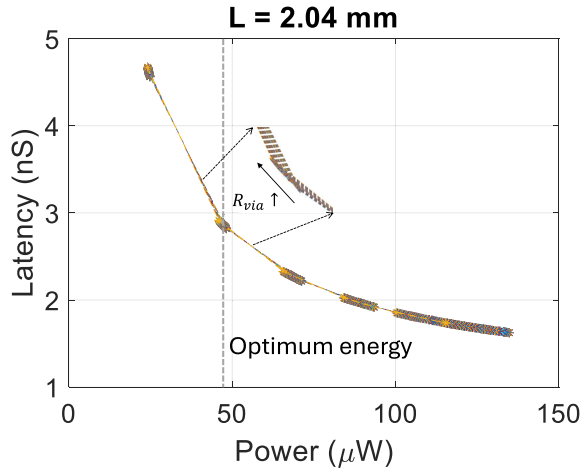


Fig. 4. Obtaining the optimum energy overhead induced by the dynamic segmented bus (DSB) loading a segment of 2.04 mm, each point in this graph refers to the specific size of the transmission gate (TG)-based switches.

TABLE III
NUMBER OF UNIQUE ACCESSED ADDRESSES IN AN LLC FOR EIGHT WORKLOADS FROM THE SPEC 2017 BENCHMARK SUITE

Workload	Unique accessed addresses [MB]	Class (number of occupied banks)
602	3.72	1
605	15.71	2
607	8.65	2
623	0.36	1
625	1.03	1
641	0.37	1
649	112.07	15
654	11.50	2

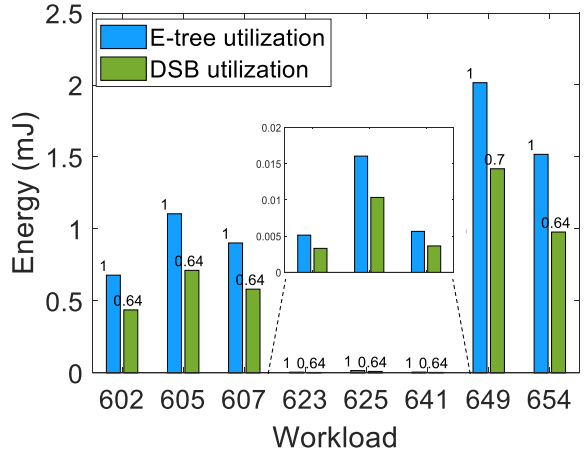


Fig. 5. Per-workload energy improvement by utilizing the DSB.

DSB overhead, we can estimate the per-workload energy for conventional E-tree and DSB, accounting for the number of accesses to each bank, as well as the distance between the corresponding bank and the root access pin.

Fig. 5 shows the per-workload energy improvement by utilizing the DSB structure as the interbank interconnect of an SRAM-based LLC. The amount of per-workload energy improvement for the workloads that only occupy the closest ring (all but 649) is around 36%. However, due to the much larger required LLC for workload 649 and subsequently activation of all the LLC rings, utilizing the DSB and mapping of the most frequently accessed addressed to the closest rings leads to a smaller energy gain of 30%.

V. CONCLUSION

Improving the energy efficiency of the VLSI technology in advanced technology nodes is challenging mainly due to the interconnect dominance. To address this issue, in this article, we have targeted a large SRAM-based last-level cache (LLC) and aimed to improve its energy efficiency via the utilization of the dynamic segmented bus (DSB) as its interbank interconnect. To adopt the DSB for an intramacro (interbank) LLC interconnect, we have also presented a workload-aware control scheme. Eventually, by optimizing the energy overhead induced by DSB, we achieve an average energy improvement of 35% across evaluated workloads.

ACKNOWLEDGMENT

The authors would like to thank Khanh Huynh (Ph.D. student at Drexel University) for sharing their segmented bus information.

REFERENCES

[1] S. Kengeri, "Heterogeneous integration in the AI era," presented at the Keynote Speech 41st IEEE VTS, 2023.
[2] H.-H. Liu et al., "Extended methodology to determine SRAM write margin in resistance-dominated technology node," *IEEE Trans. Electron Devices*, vol. 69, no. 6, pp. 3113–3117, Jun. 2022.
[3] S. Salahuddin et al., "Buried power SRAM DTCO and system-level benchmarking in N3," in *Proc. IEEE Symp. VLSI Technol.*, 2020, pp. 1–2.
[4] M. K. Gupta P. Weckx, M. P. Komalan, and J. Ryckaert, "Impact of interconnects enhancement on SRAM design beyond 5nm technology node," in *Proc. IEEE ISCAS*, 2023, pp. 1–5.
[5] M. Mayahinia, H.-H. Liu, S. Mishra, Z. Tokei, F. Catthoor, and M. Tahoori, "Electromigration-aware design technology co-optimization for SRAM in advanced technology nodes," in *Proc. IEEE DATE*, 2023, pp. 1–6.
[6] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0," in *Proc. 40th Annu. IEEE/ACM MICRO*, 2007, pp. 3–14.
[7] Z. Pei et al., "Technology/memory co-design and co-optimization using E-tree interconnect," in *Proc. GLVLSI*, 2023, pp. 159–162.
[8] J. Chen, W. B. Jone, J. S. Wang, H. I. Lu, and T. F. Chen, "Segmented bus design for low-power systems," *IEEE Trans. VLSI*, vol. 7, no. 1, pp. 25–29, Mar. 1999.
[9] J. Guo et al., "Physical design implementation of segmented buses to reduce communication energy," in *Proc. ASP-DAC*, 2006, pp. 1–6.
[10] J. Guo, A. Papanikolaou, P. Marchal, and F. Catthoor, "Energy/area/delay tradeoffs in the physical design of on-chip segmented bus architecture," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 15, no. 8, pp. 941–944, Aug. 2007.
[11] A. Balaji, Y. Wu, A. Das, F. Catthoor, and S. Schaafsma, "Exploration of segmented bus as scalable global interconnect for neuromorphic computing," in *Proc. GLVLSI*, 2019, pp. 495–499.
[12] K. Heyrman, A. Papanikolaou, F. Catthoor, P. Veelaert, K. Debosschere, and W. Philips, "Energy consumption for transport of control information on a segmented software-controlled communication architecture," in *Proc. Int. Workshop Appl. Reconfig. Comput.*, 2006, pp. 52–58.
[13] S. Song et al., "Experiments with SPEC CPU 2017: Similarity, balance, phase behavior and simpoints," Dept. Electr. Comput. Eng., LCA Group, Univ. Austin, TX, USA, Rep. TR-180515-01, 2018.
[14] S. Singh et al., "Memory centric characterization and analysis of SPEC CPU2017 suite," in *Proc. ACM/SPEC Int. Conf. Perform. Eng.*, 2019, pp. 285–292.
[15] I. Ciofi et al., "Impact of wire geometry on interconnect RC and circuit delay," *IEEE Trans. Electron Devices*, vol. 63, no. 6, pp. 2488–2496, Jun. 2016.
[16] I. Ciofi et al., "Modeling of via resistance for advanced technology nodes," *IEEE Trans. Electron Devices*, vol. 64, no. 5, pp. 2306–2313, May 2017.