

Key-value Store Chip Design for Low Power Consumption

Yuta Tokusashi, Hiroki Matsutani, and Hideharu Amano
Dept. of Information and Computer Science, Keio Univ. JAPAN
E-mail: {tokusasi, matutani}@arc.ics.keio.ac.jp, wasmii@am.ics.keio.ac.jp

Abstract

Low-power embedded systems often require simple but flexible functionality for data management. Key-value store is one of data stores, which provides simple API and is easy to scale-out. We developed a dedicated key-value store core as a low-power embedded storage. In this paper, key-value store chip is fabricated with Renesas SOTB 65-nm process technology. We evaluate the real chip in terms of power consumption and operating frequency by tuning VDD and body bias voltage. The chip achieved 11.2mW as power consumption with 0.7V VDD if the target frequency is 40MHz. When data rate is low, the system can reduce power consumption by tuning VDD and body bias voltage.

Keywords: Key-value Store, Embedded System, Low-power, Body Bias

1 Introduction

Low-power embedded systems often equip some storages for storing data and caches. For example, packet filtering rules are stored in home security systems in order to prevent specific traffic patterns. A typical approach to store such rules for fast rule access is to use Contents Addressable Memory (CAM), which typically consumes significant power. Another approach is to use key-value store in which data is represented as a key-value pair. Key-value pairs are stored in RAM. A hashed key is used as address of the RAM to access the specific key-value pair. In this paper, we focus on the key-value structure as a practical data structure in terms simplicity and generality and show key-value store chip design. The key-value store is optimized for the low power. More specifically, it can adjust the performance vs. energy efficiency by tuning VDD and body bias voltages at runtime.

CAM has been used for packet forwarding in network switches and routers. Search-lines on CAM are connected to all flip-flops on CAM, so power consumption is higher than RAM. On the other hand, the key-value store uses RAMs for hash table, and thus the power consumption is low. Regarding the key-value store acceleration, FPGA and ASIC based key-value store designs have been explored especially for data centers [1, 2, 3, 4]. These FPGA and ASIC solutions can achieve high energy efficiency, thanks to the dedicated key-value store circuit. Also, they can mitigate packet processing overheads by avoiding network protocol stack of OS kernel.

Contributions of this work are as follows. 1) Key-value store chip is designed and fabricated as a real chip. 2) The real chip is evaluated in terms of power consumption and operating frequency by tuning VDD and body bias voltage. 3) The results demonstrate that the chip can customize performance vs. energy-efficiency by tuning VDD and body bias voltage. This work shows that the real chip achieves low power consumption, which indicates that this chip can be used for embedded storages such as those for home security systems.

2 Key-value Store Chip Design

Key-value store is a widely used software system for web cache in cloud computing. Key-value store uses key-value pairs as data structure to be stored in memory. To search a specific value, a query including the corresponding key is processed, calculating a hash value based on the key and accessing the entry of the hash table based on this hash value.

Key-value store accelerators [1, 2, 3, 4] have been explored on FPGA and ASIC in order to achieve high energy efficiency. Since one of major bottlenecks on software system for key-value store is network protocol stack, key-value store accelerators offloaded to network interface card is a silver bullet for high performance in terms of throughput and latency. The implemented KVS (key-value store) chip provides energy-efficient key-value store functionality that can change VDD and body bias voltage to adjust performance vs. energy-consumption. It is assumed to run on network or stacked chips.

2.1 Entire Chip Design

The implemented chip has 32bit data interfaces as input/output data bus due to chip size limitation. As shown in Fig. 3, two key-value store cores which are denoted as a processing element (PE) are designed: the one uses DFF arrays inside a PE as data store to verify the design without SRAM, another uses SRAM to verify the design including SRAM. In addition, SRAM debug core is designed for the purpose of validation for SRAM. Read/write access is performed via I/O pins. Two bits selector is provided to choose these three modes.

To secure more storage, we are assuming to use an external memory chip which affects latency and performance, considering memory access latency. This chip has two independent PEs. To improve throughput of key-value store, possibly multiple PEs could be implemented on a chip, or multiple chips can be used by consistent hashing to distribute a query with a key. Each PE has 72bit data width interfaces: 64bit is data and 8bit is header. The header indicates information of key-, value-position and last flit. As a prototype, a PE supports 64B key and 64B value as fixed size. To support variable key/value sizes, more logic cells could be utilized.

2.2 PE Design

A PE uses cyclic redundancy check (CRC) as hash function to calculate hash value to access to a value on data store. A single SRAM cell used with a PE is divided to two regions: hash table region and data store region. For memory allocation, we used slab allocator running on MIPS R3000 compatible processor with micro code. When a SET request query is processed in PE,

a new value entry is calculated by the MIPS processor. When a DELETE query is processed in PE, the address of deleted value entry is stored in free list on DFF arrays. This software processing is bottleneck in the current design. Possible future work is developing this part with dedicated hardware. From RTL simulation, 82 clock cycles to execute GET (read request) and 169 clock cycles to execute SET (write request) are required in the key-value store PE. Since this prototype supports 64B key and 64B value as fixed size, we expect lower clock cycles in the case of home security system due to IPv4/IPv6 address size (e.g., 4B/16B). Our prior work [5] shows that 5 PEs interconnected with a crossbar switch achieved 10GbE line rate. Note that, since the design of [5] runs on 200MHz, more PEs are required to achieve the line rate using this case (e.g., 40-65MHz).

3 Chip Implementation

KVS chip was implemented with Renesas SOTB (Silicon On Thin Box) 65-nm process technology. Tools used for the design are Cadence ncvrilog (version 15.10-s008) for RTL Simulation, Synopsys Design Compiler (version I-2013.12-SP2) for synthesis and Synopsys IC Compiler (version I-2013.12-ICC-SP2) for layout. SOTB [6] is a novel Fully Depleted Silicon on Insulator (FD-SOI) technology. Transistors are constructed on an ultra thin BOX layer to suppress the detrimental Short Channel Effect (SCE) and the V_{TH} variation caused by Random Dopant Fluctuations (RDF). In SOTB, the body bias can be used to optimize the performance and power consumption after fabrication. For an nMOS (pMOS) transistor, V_{BN} (V_{BP}) is given to its p-well (n-well). Zero bias, where V_{BN} and V_{BP} are equal to the source voltage (That is, $V_{BN} = 0$, $V_{BP} = V_{DD}$), means that the transistor works with its normal V_{TH} . When the forward bias ($V_{BN} > 0$ and $V_{BP} < V_{DD}$) is given, V_{TH} is decreased, and the leakage current is increased, while the operational speed is enhanced.

As shown in Fig. 1, the chip was implemented on a $3\text{mm} \times 3\text{mm}$ die. Fig. 2 shows the micro-photograph of the chip. Only a square corresponding to the core can be observed.

4 Real Chip Evaluation

The advantage of the SOTB process is its low power consumption with a low supply voltage (V_{DD}). Fig. 4 shows the relationship between power consumption, operational frequency and supply voltage. The area without values means that the chip was not operational with the given supply voltage, and the area shaded with green shows that the chip was operational with the zero bias. If the target frequency is 40MHz, the power consumption was only 11.2mW with 0.7V V_{DD} . Although the maximum frequency was 55MHz with the zero bias, we can enhance it by giving the forward bias ($V_{BN} > 0$). Here, we used the balanced body bias, that is ($V_{BN} + V_{BP} = V_{DD}$), thus, V_{BP} is automatically fixed when V_{BN} is fixed. We only show the V_{BN} here. The maximum operational frequency can be enhanced to 60MHz with $V_{BN} = 0.6V$ and 65MHz with $V_{BN} = 0.9V$. As shown in Fig. 4, the power consumption was not increased so much. Tuning VDD and controlling body-bias can improve performance when throughput is required, as shown in Fig. 5. Fig. 6 shows the relationship between the leakage current and V_{BN} . Unlike the original SOTB process[6], recent design kit focuses on suppressing the leakage current with the moderate performance gain with the forward biasing.

5 Conclusions

Key-value store chip was designed and implemented for low power embedded systems such as home security systems. Although typical CAM as a solution to hold forwarding rules typically consumes significant power, the implemented key-value store chip achieved low power consumption. Results indicate when input data rate is high on the target system, operating frequency can be increased by tuning VDD and body bias voltage to improve its performance. When input data rate is low, the system can reduce power consumption by decreasing VDD and tuning body bias voltage.

We will expect to explore the practical use case with more memory space. The system assures more memory space by introducing stacked memory chip. Possible future direction is implementation with Through-Chip Interface (TCI) and then KVS chip can utilize wider memory space with memory chips via TCI.

Acknowledgment

This work was supported by JSPS KAKENHI S Grant Number 25220002. The device model of SOTB in this study has been provided by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Renesas Electronics Corp. The authors also thank Synopsys Inc. for the EDA tools support.

References

- [1] Tokusashi, Y. and Matsutani, H.: A Multilevel NoSQL Cache Design Combining In-NIC and In-Kernel Caches, *Hot Interconnects* (2016).
- [2] Tokusashi, Y. and Matsutani, H.: Multilevel NoSQL Cache Combining In-NIC and In-Kernel Approaches, *IEEE Micro*, Vol. 37, No. 5, pp. 44–51 (2017).
- [3] Lim, K., Meisner, D., Saidi, A. G., Ranganathan, P. and Wenisch, T. F.: Thin Servers with Smart Pipes: Designing SoC Accelerators for Memcached, *Proceedings of the International Symposium on Computer Architecture (ISCA'13)*, pp. 36–47 (2013).
- [4] Chalamalasetti, S. R., Lim, K., Wright, M., AuYoung, A., Ranganathan, P. and Margala, M.: An FPGA Memcached Appliance, *Proceedings of the International Symposium on Field Programmable Gate Arrays (FPGA'13)*, pp. 245–254 (2013).
- [5] Tokusashi, Y., Matsutani, H. and Zilberman, N.: LaKe: The Power of In-Network Computing, *Proceedings of the International Conference on ReConfigurable Computing and FPGAs (ReConFig2018)* (2018).
- [6] Y. Morita et al.: Smallest V_{TH} Variability Achieved by Intrinsic Silicon on Thin BOX (SOTB) CMOS with Single Metal Gate, *Proceedings of 2008 Synposia on VLSI Technology and Circuits*, pp. 166–167 (2008).

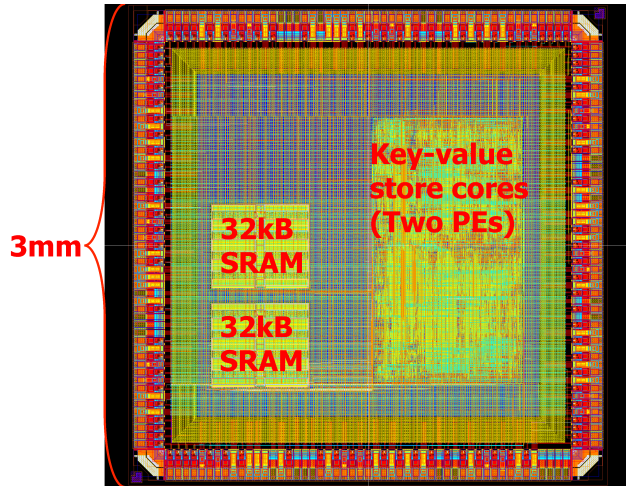


Figure 1: Layout of KVS chip

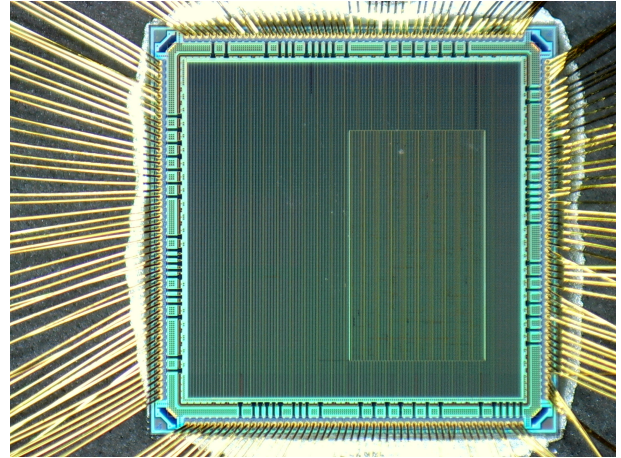


Figure 2: Photo of the KVS chip

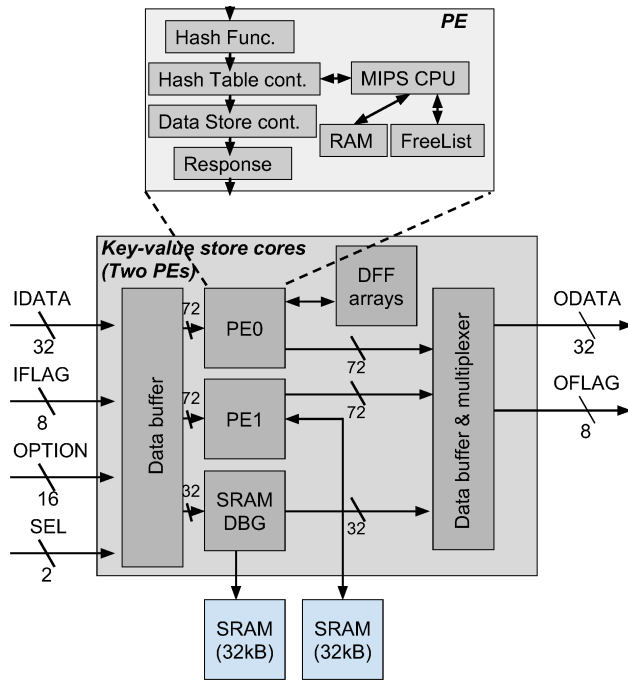


Figure 3: Block diagram of the KVS chip

P(mW)	f(MHz)					
VDD(V)	40	45	50	55	60	65
0.9	18.9	21.6	24.3	24.3	30.6	33.3
0.85	17	19.55	21.25			
0.8	15.2	16.8	18.4			
0.75	12.75	15.75				
0.7	11.2					
	VBN=0.0			VBN=0.6 VBN=0.9		

Figure 4: Total power consumption of KVS chip

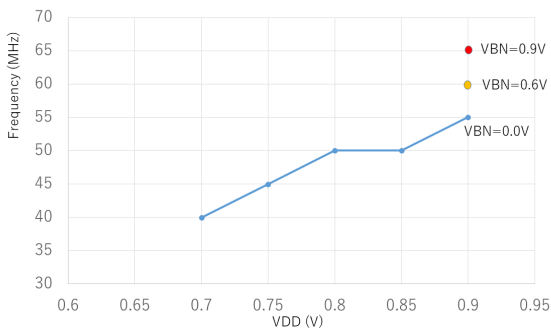


Figure 5: Operational frequency (MHz) versus VDD (V)

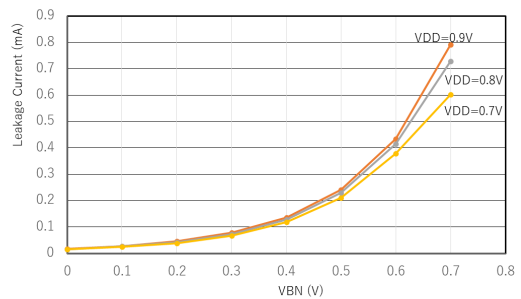


Figure 6: Leakage current (mA) versus VBN (V)