Autonomous high-speed serial link power management depending on required link performance for HMC

D.-I. Jeon and K.-S. Chung✉

Many studies on 3D-stacked dynamic RAMs (DRAMs) have been conducted to overcome the shortcomings of conventional DRAM. The hybrid memory cube (HMC) is one of the most promising 3D-stacked DRAMs, thanks to its high bandwidth and expandable structure. However, a high-speed serial link that interfaces the CPU and HMC consumes significant power, primarily because of the high overhead incurred in synchronising its clock. Although the link provides low-power modes, managing them is very difficult because of their long mode transition times. An autonomous power management method for the high-speed link is proposed. The proposed method determines the optimal number of active links while satisfying the required link performance. Simulations demonstrate that the proposed method reduces link power consumption by an average of 63.06% with a performance degradation of only 1.36%. Therefore, this proposed autonomous link power management is an outstanding option for low-power HMC-based systems.

Introduction: The hybrid memory cube (HMC) is structurally different from the conventional dynamic RAM architecture. Memory commands and data between the CPU and the HMC are transmitted in both directions across a high-speed serial link called the serialiser/deserialiser (SerDes). Data to be transmitted are serialised and transmitted across the link in a bit-serial form on each lane and then are reassembled at the receiving end of the link. To transmit data on a lane, a clock signal is embedded in the data stream itself, as opposed to general interfaces, where clock and data signals are separate. When an HMC is initially powered on, the link and protocol initialisation called link training, which includes link clock synchronisation, is followed. After initialisation, the link must continue transmission to maintain the clock synchronisation; even when no data are to be transmitted, the link must transmit NULL packets. Otherwise, link retraining should be repeatedly processed. This implies that the link consumes a significant amount of power. Ahn *et al.* [1] found that it would account for about 73% of the energy dissipated by a conventional HMC-based system.

According to HMC specification [2], each link can be independently set to a low-power mode such as a sleep or down mode. A sleep mode disables the link's SerDes circuitry, while a down mode consumes less power because it disables both the SerDes circuitry and the link's PLLs. To return from low-power modes to normal, active mode operation, a link retraining mode should be initiated. Every mode transition requires meeting its own timing parameters as specified in the HMC specification. The mode transition incurs significant performance overhead due to its long transition time. For example, the transition time from the sleep mode to the active mode is 1055.5 μs according to HMC specification [2]. Therefore, managing the link's low-power mode to achieve the required link performance is critical.

Related works: Ahn *et al.* [1] proposed a dynamic power management method for off-chip links based on a delay monitor. This method partially disables off-chip links of an HMC in an adaptive manner. The main idea is to periodically estimate the minimum number of links to achieve a certain average link delay. We name this management method 'delay monitor management (DMM)' here. DMM approach does not take real HMC characteristics into consideration since the actual timing parameters of the mode transition in accordance with the HMC specification are not considered. In addition, the measured average link delay is defined by the burst length of a request and the signalling rate of a link, even though all communication across the link is packetised. Therefore, DMM may not be able to properly manage the link mode due to inaccurate estimation of performance degradation.

It is very important to determine an adequate number of active links when considering the trade-off between power consumption and performance. The method of proposed autonomous link power management determines the optimal number of active links by two metrics: the cache miss status holding registers (MSHRs) and the link monitor. The proposed method requires a slight modification to the conventional HMC architecture and incurs little performance degradation while reducing power consumption.

Miss SHRs: If a cache miss occurs, the processor will stall until the outstanding cache miss is handled. A non-blocking cache allows the processor to continue to perform useful work even in the presence of cache misses as long as dependency constraints are not observed. To employ a non-blocking cache, Kroft [3] proposed MSHRs to hold information about outstanding misses. Each MSHR includes a valid bit, a data block physical address, a destination register number, and so on. An MSHR is allocated when a cache miss occurs, and the information regarding the cache miss is stored in the allocated MSHRs. Many misses in MSHRs implies that memory transactions are committed actively. Therefore, the number of allocated MSHRs may be strongly correlated to the required memory bandwidth in the near future. To confirm the correlation between the number of allocated MSHRs and the HMC bandwidth, experiments were conducted by the implementation of gem5 [4] and CasHMC [5] simulators. The number of MSHRs and HMC bandwidth were measured on average every 1 ms. Fig. 1 shows the experimental results with respect to two programs in the SPEC CPU2006 benchmark: *gobmk* and *perlbench* [6]. As shown in this figure, the HMC bandwidth was closely related to the number of MSHRs. In particular, the increasing and decreasing tendencies of MSHRs are closely correlated to the bandwidth. Therefore, the HMC bandwidth was predicted by the number of allocated MSHRs.

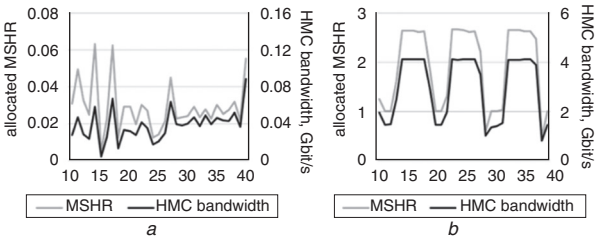

Fig. 1 *Correlation between number of the allocated MSHRs and HMC bandwidth*

a gobmk
b perlbench

Link monitor: Memory transactions are mainly generated by the main processor, but other logic blocks such as graphic PUs and direct memory access inputs/outputs issue memory requests as well. Since the number of MSHRs indicates memory requests only from the main processor, we need to determine the total actual bandwidth of the link. Thus, we propose a new type of hardware called the link monitor to measure actual link bandwidth.

All HMC in-band communications across a link are packetised. Each packet field is described in the HMC specification. The field length (LNG) indicates the total number of flits in the corresponding packet. As a flit is a 16 B flow control unit, multiplying LNG by 16 gives the packet length. As a result, the total length of transmitted packets can be derived by accumulating every LNG value. The proposed method accumulates the lengths of transmitted packets in each user-defined epoch. The request and response LNG accumulators are separated because links employ unidirectional communication. Thus, the downstream link bandwidth (DB) and the upstream link bandwidth (UB) after one epoch has elapsed are derived.

Autonomous link power management: Link power management should carefully perform low-power mode (sleep or down mode) transitions while taking the required link bandwidth into account. The number of MSHRs implies the necessary memory bandwidth in the near future, but considers memory requests from only the CPU. In contrast, the link monitor measures the actual link bandwidth through the transmitted packets. However, the link bandwidth was measured during the last epoch, and the required bandwidth for the current epoch may be different. Even though the two metrics have respective shortcomings, they supplement each other well.

Fig. 2 shows the structure of proposed link power management. A link power manager handles link mode transitions, and decides how many links should remain in active mode or wake up to the link retraining mode in every epoch in accordance with MSHRs and the link monitor. When MSHR is allocated or deallocated, the cache controller delivers the updated number of MSHRs to the link power manager. Thus, the number of MSHRs on the average can be derived by the

link power manager in every epoch. The link monitor extracts LNG field from all the transmitted requests and response packets across the link. Extracted LNG values are accumulated in the link power manager to derive the DB and UB. Finally, the link power manager determines the number of active links in accordance with Table 1. Since the performance bottleneck due to the link low-power mode is a crucial problem, the number of active links is chosen by the higher value between the MSHR and the link monitor. Here, α and β are user-defined scaling factors to manage the trade-off between power consumption and performance for MSHR and link monitor, respectively. When a scaling factor becomes bigger, the link power manager may increase the number of active links for the upcoming epoch in favour of improved performance. If the obtained number of active links is different from the current number of active links, the link power manager conducts a link mode transition. The link low-power mode transitions and the related timing parameters strictly follow the conventional HMC.

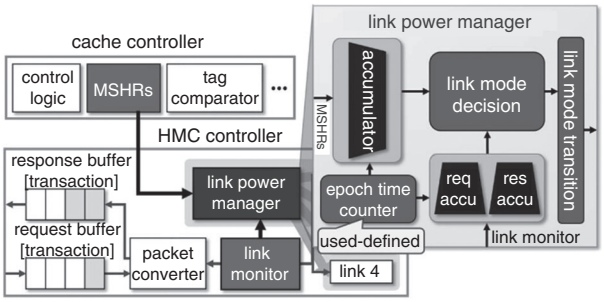

Fig. 2 *Structure of autonomous link power management*

Table 1: Number of active links with respect to MSHR and link bandwidth

Active link	MSHRs	Link monitor
4	4≤αMSHR	3xLB≤βDB or 3xLB≤βUB
3	3≤αMSHR<4	2xLB≤βDB<3xLB or 2xLB≤βUB<3xLB
2	2≤αMSHR<3	LB≤βDB<2xLB or LB≤βUB<2xLB
1	αMSHR<2	βDB<LB or βUB<LB

Note: α is scaling factor for MSHR and β is scaling factor for link monitor.
LB is the maximum one link bandwidth with respect to link configuration (LB = link width×link speed/8).

Experiments: To verify the proposed method, a simulation environment was designed by combining two simulators: gem5 [4] and CasHMC [5]. To validate the simulation accuracy, various SPEC CPU2006 benchmarks [6] were executed. Energy consumption of a link is modelled as 5 mW/Gbit/s according to OmniPHY, that is, one of HMC consortium members. The power consumption in the sleep mode is about 10% of it, and that in the down mode is about 1%. The scaling factors for MSHR (α) and link monitor (β) were set to 1.0 and 1.0, respectively. The user-defined epoch was set to 1 ms.

Fig. 3 shows the simulated results in terms of the average memory request latency and the link power consumption with regard to four link power management methods: DMM (period = 100 µs, $\alpha = 0.05$), MSHRs only, the link monitor only, and the proposed method. The simulation results were normalised by the results where no link power management was employed. Obviously, all link power management methods consume much lower link power than no link management. However, DMM has a significant increase in the memory request latency because of naïve monitoring. On the contrary, in the case of MSHRs and the link monitor, the memory request latency was increased slightly, but the link power consumption was reduced by up to 67.19%. The proposed autonomous link power management employs both MSHRs and link monitor and conducts a link mode transition with minimal impact on the performance degradation. The simulation results show 63.06% reduction with no management in the link power consumption with only a 1.36% performance degradation on average.

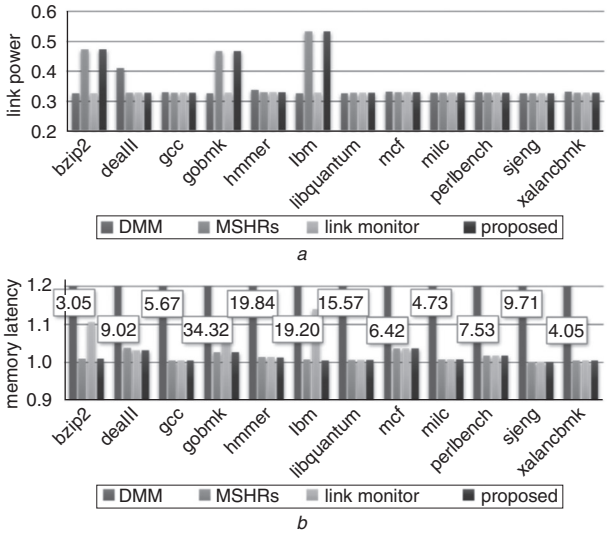

Fig. 3 *Simulation results normalised by no link power management with regard to four link power management methods*

a Link power consumption
b Average memory request latency

Conclusion: The proposed method refers to the number of the allocated MSHRs and the measured link bandwidth by a link monitor and then decides the optimal number of links in terms of minimising power consumption. Even though the proposed autonomous management method employed a simple monitoring technique with a smaller overhead, the performance is degraded only slightly. Therefore, the proposed autonomous link power management is claimed to be very effective with respect to both power consumption and performance.

Acknowledgment: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A09061079).

Submitted: *26 March 2018*
doi: 10.1049/el.2018.0997
One or more of the Figures in this Letter are available in colour online.

D.-I. Jeon and K.-S. Chung (*Department of Electronics and Computer Engineering, Hanyang University, Seoul, Republic of Korea*)

✉ E-mail: kchung@hanyang.ac.kr

References

1 Ahn, J., Hong, S., and Choi, K.: 'Dynamic power management of off-chip links for hybrid memory cubes'. Design Automation Conf. (DAC), San Francisco, USA, June 2014
2 Hybrid Memory Cube Consortium (HMCC): 'Hybrid Memory Cube Specification 2.1'
3 Kroft, D.: 'Lockup-free instruction fetch/prefetch cache organization'. Int. Symp. Computer Architecture (ISCA), Minneapolis, USA, May 1981
4 Binkert, N., Beckmann, B., Black, G., *et al.*: 'The gem5 simulator', *ACM SIGARCH Comput. Arch. News*, 2011, **39**, (2), pp. 1–7
5 Jeon, D.I., and Chung, K.S.: 'CasHMC: a cycle-accurate simulator for hybrid memory cube', *Comput. Arch. Lett.*, 2017, **16**, (1), pp. 10–13
6 Henning, J.L.: 'SPEC CPU2006 benchmark descriptions', *ACM SIGARCH Comput. Arch. News*, 2006, **34**, (4), pp. 1–17