A Customized Design of DRAM Controller for On-Chip 3D DRAM Stacking

Tao Zhang[1], Kui Wang[2], Yi Feng[3], Xiaodi Song[2], Lian Duan[1],
Yuan Xie[1], Xu Cheng[3], and Youn-Long Lin[4]

[1]Department of Computer Science and Engineering, The Pennsylvania State University, PA 16802, USA
[2]Peking University Unity Microsystems Technology Co. Ltd, Beijing, China
[3]Department of Computer Science, Peking University, Beijing, China
[4]Computer Science Department, National Tsing Hua University, Hsinchu, Taiwan

***Abstract*—To address the "memory wall" challenge, on-chip memory stacking has been proposed as a promising solution. The stacking memory adopts three-dimensional (3D) IC technology, which leverages through-silicon-vias (TSVs) to connect layers, to dramatically reduce the access latency and improve the bandwidth without the constraint of I/O pins. To demonstrate the feasibility of 3D memory stacking, this paper introduces a customized 3D Double-Data-Rate (DDR) SDRAM controller design, which communicates with DRAM layers by TSVs. In addition, we propose a *parallel access policy* to further improve the performance. The 3D DDR controller is integrated in a 3D stacking System-on-Chip (SoC) architecture, where a high-bandwidth 3D DRAM chip is stacked on the top. The 3D SoC is divided into two logic layers with each having an area of $2.5 \times 5.0mm^2$, with a 3-layer 2Gb DRAM stacking. The whole chip has been fabricated in Chartered 130nm low-power process and Tezzaron's 3D bonding technology. The simulation result shows that the on-chip DRAM controller can run as fast as 133MHz and provide 4.25GB/s data bandwidth in a single channel and 8.5GB/s with parallel access policy.** [1]

I. Introduction

As technology scales, 3D IC technology has been proposed as a valuable solution in current and future chip designs to extend Moore's Law. With dense TSVs, 3D ICs can offer us: (1) wire length reduction; (2) low latency; (3) high data bandwidth; and (4) heterogenous design [1]. Specifically, 3D IC allows us to stack main memory on the logic chip to reduce the memory access latency and improve the memory bandwidth. Moreover, moving the off-chip memory into the chip can eliminate the I/O pin limitation which further improves the memory performance. The memory stacking, however, enforces people rethinking the interface design between the logic and memory. Therefore, a new 3D memory controller needs to be considered to make better use of 3D on-chip memory.

There exist a few work related to 3D on-chip memory stacking. Loh proposed several aggressive DRAM organizations that are stacked on a multi-core processer [2]. Both pseudo-3D DRAM and true-3D DRAM organizations are exploited to take advantage of 3D stacking. Saito et al. implemented a 3D SoC, in which the SRAM is stacked on the logic chip [3]. The SRAM is reconfigurable so that the memory space can be reallocated due to different SoC architectures. Woo et al. reorganized the memory hierarchy by implementing a wider memorybus with plenty of TSVs [4]. As a result, 64 memory accesses can be processed in parallel between the last level cache (LLC) and the 3D DRAM. Kim et al. tried to improve the memory performance with multiple memory controllers (channels) [5]. Four memory controllers are introduced to increase the bandwidth with little coordination between each other. To our best knowledge, however, most of prior work fall into software simulations rather than hardware implementations even though the 3D DRAM has been silicon-proven by the industry. In contrast to previous work, our goal is to implement a 3D DRAM controller integrated into a 3D SoC to demonstrate the feasibility of 3D on-chip DRAM stacking.

The rest of paper is organized as follows. Section II briefly introduces our proposed 3D SoC architecture and the 3D DRAM used in this project. Section III presents the structure of the 3D DDR controller and the simulation result. Section IV shows the implementation of 3D DRAM stacking and Section V concludes this paper.

II. 3D SoC Architecture And 3D DRAM

In this section, we first introduce the 3D SoC architecture in which the 3D DDR controller is used. We then describe the 3D DRAM memory which is stacked on top of the SoC logic chip.

A. 3D SoC Architecture

To support real-time multimedia applications with H.264 encoding, a 3D SoC architecture is proposed as our platform. As shown in Fig. 1, AMBA AHB is employed as system bus [6]. USB On-The-Go controller, H.264 encoder, and the dedicated 3D DDR controller are also integrated into this SoC architecture. In addition, we acquire a RISC processor UniCore-II as the computing unit. This SoC is mapped into a two-layer 3D architecture, which is shown in Section IV.

B. 3D DRAM

The 3D DRAM chip used in this work is a state-of-the-art product from Tezzaron [7]. To achieve both high performance

[1]Zhang and Xie are supported in part by NSF 0903432, 0643902 and 0702617, and a SRC grant.

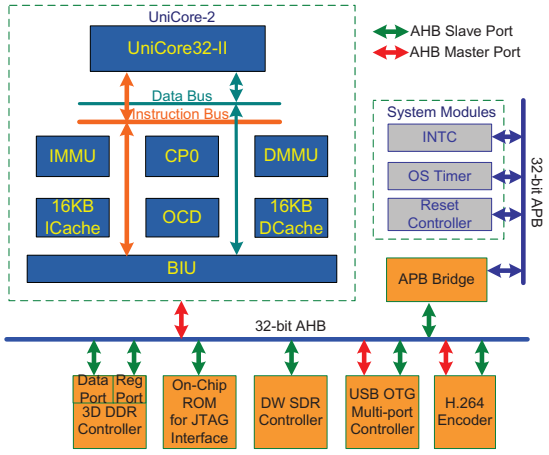


Fig. 1. Schematic view of proposed 3D SoC

TABLE I
PARAMETER OF 3D DDR CHIP

Num. of Layers	3
Total Capacity	2Gbits (256MB)
Clock Frequency	1GHz (Max.)
Refresh Mode	Automatic
Refresh Rate	64ms
Num. of Data Channel	8
Data Width Per Channel	128bits
Num. of Pins Per Channel	294
Burst Length	4 or 8
Addressing Mode	Sequential

and high data density, the DRAM chip is separated into peripheral layer and cell layer, where each layer is optimized respectively. The total capacity of this 3D DRAM is 2Gb (256MB) with 1Gb on each cell layer. Eight data channels are employed to attain high bandwidth. In addition, one MailBox channel is used to initialize the DRAM when the system is powered on. The data width of each channel is 128-bit. To provide huge bandwidth, every data channel adopts the DDR protocol that can return 256-bit data in a single memory cycle. Each data channel is actually independent with the others, which means all of channels can be accessed in parallel.

The 3D DRAM is composed of eight banks which has a one-to-one relation with the data channels. The bank and the data channel are therefore interchangeable in this paper. Similar to the conventional DRAM technology, the 3D DRAM also has 5 main operations known as *precharge*, *refresh*, *row address*, *column read* and *column write*. The 3D DRAM can run as fast as 1GHz with 64ms refresh rate. Table I lists the basic parameters of this 3D DRAM chip. As shown in line 8, as many as 294 pins are used for each data channel so that the DDR controller has more than 2,300 pins in total. Without 3D on-chip stacking, it's impractical to afford so many I/O pins in conventional 2D topology according to the constraints of power, area, and cost.

III. 3D DDR CONTROLLER

In this section, we first depict the structure of 3D DDR controller. Then, we discuss the parallel access policy in the second subsection, following the illustration of hardware supports for the novel access policy. Finally, we list the simulation results on bandwidth and area.

A. Structure of DRAM Controller

As shown in Fig.2, our DDR controller consists of three functional blocks: AHB wrapper, asynchronous FIFO (Address FIFO and Data FIFO), and DRAM wrapper. Both AHB wrapper and DRAM wrapper are controlled by finite state machines (FSM). The transition of the FSM in AHB (DDR) wrapper is triggered by AHB (DDR) clock. The usage of AHB wrapper is twofold. Firstly, the wrapper translates AHB transactions into intermediate transactions which can be recognized by DRAM wrapper, and then sends those transactions into the asynchronous FIFO. Also, it maintains the control registers to initialize the DRAM. When the SoC is powered on, UniCore-II initiates the initialization commands to the DRAM through the AHB bus and the MailBox channel, by overwriting the control registers. Therefore, two AHB slave interfaces are deployed to support the data delivery and the register configuration respectively. On the other hand, since the maximum burst length on AHB is 16, the memory burst length is set to be four to meet the longest AHB burst.

The address FIFO stores the starting address and the transaction information, such as the read/write direction, AHB burst length, and so on. The data FIFO contains both write FIFO and read FIFO. We double the write FIFO size to support the parallel access that has details in the next section. The transaction on the top of FIFO will be fetched into DDR wrapper once it is available. According to the runtime environment and the memory access pattern, the DDR wrapper may issue the command immediately or hold it until the DRAM is accessible. For instance, DRAM wrapper has to hold the command if DRAM enters *refresh* stage.

For read transactions, once DRAM wrapper successfully receives the read data from DRAM, it pushes the data into read FIFO. An AHB slave response along with the data is then generated by AHB wrapper. To simplify the implementation, neither *reply* nor *split* response is applied to the controller. On the other hand, all data channels need to refresh the row data in DRAM simultaneously. A refresh unit is employed to generate refresh requests periodically. Once DRAM wrapper detects the refresh request, it should terminate the current access and enforce all the channels going to the refresh phase. Sometimes, the AHB transaction may cross a 512b-aligned address boundary (one DRAM burst) so that DRAM wrapper needs to split the original transaction into two separate accesses.

B. Parallel Access Policy

From our observation, AHB bus can only have at most two outstanding transactions at the same time. Due to this limitation, even though the DRAM has eight independent channels, a two-channel *parallel access policy* is sufficient to improve the performance. We categorize those back-to-back

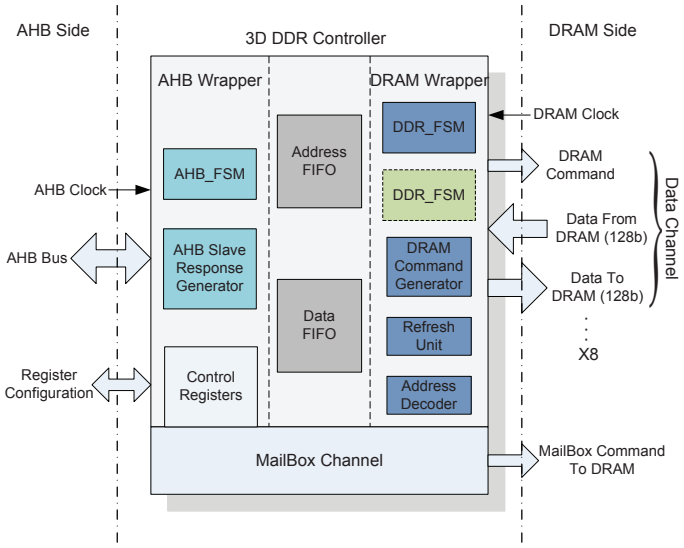


Fig. 2. Block view of 3D DRAM controller

AHB transactions as: Read-After-Read (RAR), Read-After-Write (RAW), Write-After-Write (WAW), and Write-After-Read (WAR). Based on our understanding, except for WAR, we have the opportunity to optimize the first three patterns by allowing the second outstanding transaction to be detected and processed without any stall, if they have different bank requests. In other words, under certain conditions, the DRAM controller does not necessarily wait for the completion of the first memory access but can initiate the second access command to the 3D DRAM immediately. For example, Fig.3 shows the case of RAR, where the second read follows the first read. As shown in Fig.3.(a), the DDR controller originally can only process each transaction in sequence even if the second read requests the data from a different bank. As a result, the second read needs to stall for a few cycles before it can be served. In contrast, with the parallel access policy, the second transaction can be processed immediately if it references a different data bank as shown in Fig.3.(b).

Another DRAM FSM is replicated to allow the second transaction to be processed. After AHB wrapper detects the second transaction, it informs DDR wrapper that the second access request is coming. DDR wrapper then wakes up the replica to respond to the second memory access. Sometimes, the first access may suffer a row miss while the second access enjoys a row hit. In this case, the second access will be retired prior to the first one. Considering the in-order property of AHB protocol, the DDR controller should hold the second result until the first access finishes.

C. Hardware Support

Along with these two DDR FSMs, extra hardware devices are also introduced in DDR wrapper to enable parallel accesses. An 8-bit register OpenRow is added to distinguish whether there is an open row in the bank or not. The corresponding bit will be set when a *row address* is performed to open a row and reset if a *precharge* is conducted to close the active row. Two 32-bit address registers are used to latch the

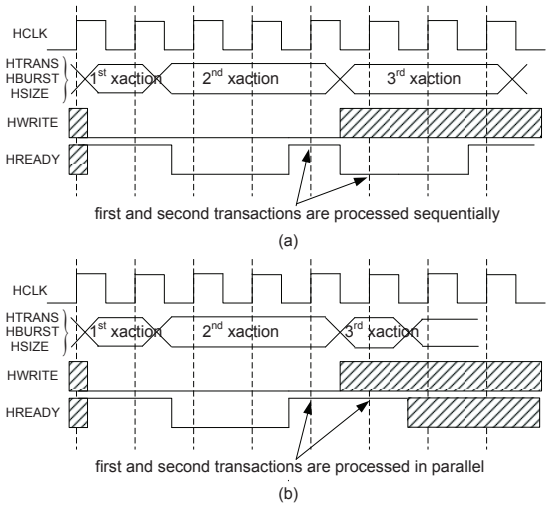


Fig. 3. Performance improvement with parallel accessing policy

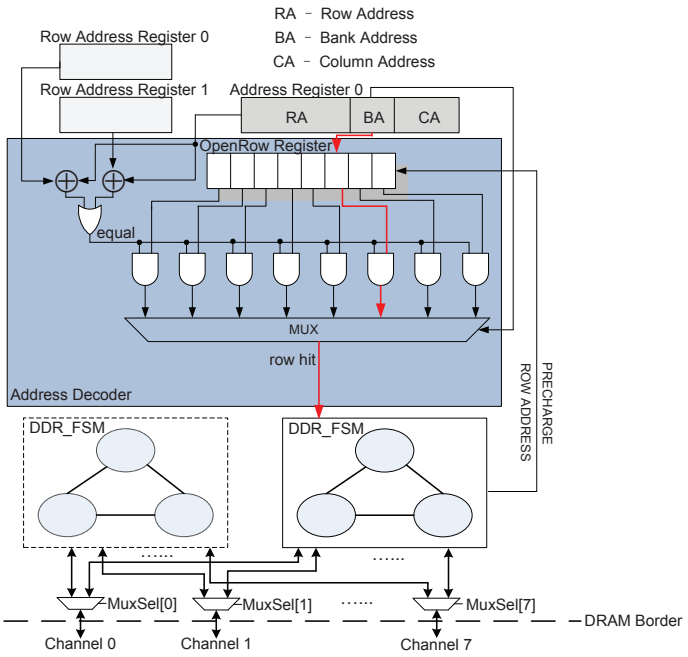


Fig. 4. Hardware support for parallel access

access addresses. Two Row Address registers are also used to record the two recent activated row addresses. Working with Row Address and OpenRow registers, Address Decoder can generate a row hit or miss to determine the state transition in DDR FSM (as shown in Fig.4, Bank5 is selected as an example). Furthermore, the register MuxSel controls the select signal for each channel's I/O multiplexer. The corresponding bit will be set to '0' when one channel is occupied by the first DDR FSM and '1' if the second FSM uses it. Finally, as mentioned above, the write FIFO size is doubled so that it can accept two longest write bursts on the AHB bus.

D. Simulation Result

We evaluate the performance and the area of DDR controller by Design Compiler. Table II shows the simulation result. The DDR controller consumes $0.78mm^2$ and can run as fast

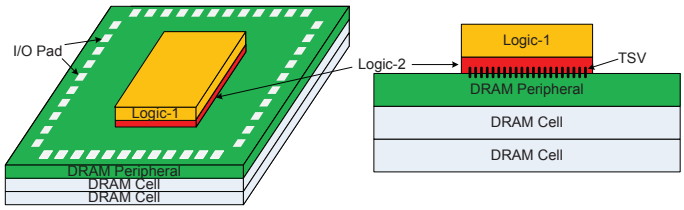


Fig. 5. 3D DRAM stacking

as 133MHz, which can be translated to 4.25GB/s bandwidth without any optimization. With the parallel access policy, the channel utilization is improved and the peak bandwidth can be doubled to 8.5GB/s. Obviously, the bandwidth can be further elevated if some approaches, such as implementing multiple memory controllers, can be used to further leverage channel independence.

TABLE II
SIMULATION RESULT

Area	0.78mm^2
DDR Clock Freq.	133MHz
Data Bandwidth	4.25GB/s (w/o Parallel Access) 8.5GB/s (with Parallel Access)

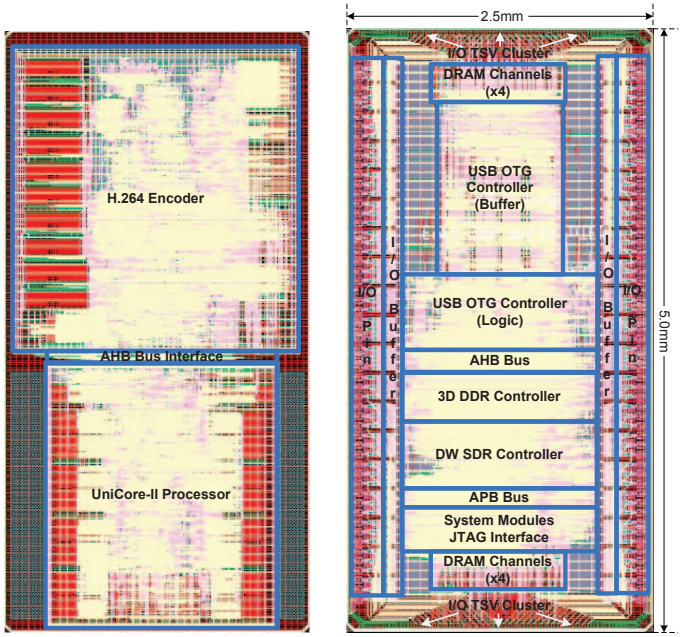
IV. 3D DRAM STACKING

The prototype chip has been fabricated in Chartered 130nm low-power process. The supply voltage of this chip is 1.5V. Fig.5 gives the overview of 3D DRAM stacking and Fig.6 shows the microphotographs of logic layers. Logic-1 layer is on the top and Logic-2 layer lies in the middle. Both logic layers are sized by $2.5 \times 5.0mm^2$ while the DRAM layers are $12.3 \times 21.8mm^2$. Logic-2 is thinner than other layers because the silicon substrate of this layer is thinned to expose TSVs.

Stacking on the 3D DRAM chip, the DDR controller can communicate with DRAM via eight data channels. As shown in Fig.6.(b), these DDR channels are divided into two groups and placed on the top and the bottom of Logic-2 layer, respectively. To achieve high reliability, multiple TSVs are bonded together to form a TSV cluster for signal delivery. Two types of TSV clusters are employed to deliver the data as well as power. The DRAM cluster is used as the carrier of DRAM signals. The I/O cluster connects the I/O pins on Logic-2 layer to the I/O pads on DRAM layer. Each I/O cluster contains 26 TSVs while each DRAM cluster contains 10 TSVs. Table III lists the physical characteristics and the total number of TSVs that we use in this work. Moreover, to address the issue of fabrication quality, bunches of dummy TSVs are also placed on both logic layers.

TABLE III
TSV PARAMETERS

Diameter	1.2μm
Pitch	4.0μm
Depth	6.0μm
Num. per DRAM Cluster	10
Num. per I/O Cluster	26
Total Number	28,908



(a) Logic-1 (b) Logic-2

Fig. 6. Microphotograph of logic layers

V. CONCLUSION

In this paper, we described the implementation of a dedicated 3D on-chip memory controller and the approach to stack memory on the logic chip. We demonstrate the feasibility of 3D memory stacking by building a 3D SoC for multimedia applications that can leverage the high memory bandwidth offered by 3D integration. The DRAM memory stacking mitigates the I/O pin limitation so that it is possible to support as many as 8 independent channels. In addition, to make use of multiple channels, we develop the parallel access policy which allows two access requests to be processed in parallel through two channels. The SoC chip has been fabricated with Chartered 130nm technology and Tezzaron's 3D bonding technology.

REFERENCES

[1] Y. Xie, J. Cong, and S. Sapatnekar, *Three-Dimensional Integrated Circuit Design: EDA, Design and Microarchitecture*. Springer, 2010.
[2] G. H. Loh, "3D-Stacked Memory Architectures for Multi-core Processors," in *Proceedings of the International Symposium on Computer Architecture*, June 2008, pp. 453–464.
[3] H. Saito, M. Nakajima, T. Okamoto *et al.*, "A Chip-Stacked Memory for On-Chip SRAM-Rich SoCs and Processors," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 15–22, January 2010.
[4] D. H. Woo, N. H. Seong, D. L. Lewis, and H.-H. S. Lee, "An Optimized 3D-Stacked Memory Architecture by Exploiting Excessive, High-Density TSV Bandwidth," in *Proceedings of the International Conference on High Performance Computer Architecture*, January 2010.
[5] Y. Kim, D. Han, O. Mutlu, and M. Harchol-Balter, "ATLAS: A Scalable and High-performance Scheduling Algorithm for Multiple Memory Controllers," in *Proceedings of the International Conference on High Performance Computer Architecture*, January 2010.
[6] ARM, "AMBA Specification," May 1999.
[7] Tezzaron, "Preliminary Specification for 8-port Memory," 2010.
[8] International Technology Roadmap for Semiconductors, 2009.
[9] G. H. Loh, "Extending the Effectiveness of 3D-Stacked DRAM Caches with an Adaptive Multi-Queue Policy," in *Proceedings of the International Symposium on Microarchitecture*, December 2009, pp. 201–212.