3D Shared Bus Architecture Using Inductive Coupling Interconnect

Akio Nomura[1], Yu Fujita[1], Hiroki Matsutani[2], and Hideharu Amano[1]

Keio University
3-14-1 Hiyoshi, Kohoku-ku,
Yokohama, Japan
wasmii@am.ics.keio.ac.jp[1] matutani@arc.ics.keio.ac.jp[2]

Abstract—**Attention has been focused on 3D chip stacking for reducing the semiconductor area per chip while maintaining the overall performance. By using inductive coupling wireless Thru-Chip Interface (TCI) in chip stack for a 3D multiprocessor, the replacement/addition/removing of chips is made possible using the inductive coupling wireless TCI in the chip stack for a 3D multiprocessor and thus, high flexibility is provided.**

A bus can be easily formed with the TCI by stacking duplex wired coils in the same place on the stacked chip. However, traditional static time division multiple access (STDMA) cannot make use of the potential bus bandwidth, while dynamic time division multiple access (DTDMA) requires a lot of coils that cannot be efficiently used for the control signals. We propose an asynchronous TDMA bus (A-TDMA bus) that uses the CSMA/CD protocol and a resonant synchronous TDMA bus (RS-TDMA bus) that uses a resonant synchronized clock and a look-ahead technique to improve the use of the bandwidth of a 3D shared bus.

The results of a network simulation using the GEM5 simulator showed that the minimum-load latency of both proposed methods was reduced by 29% in a four-chip stack and 50% in an eight-chip stack compared to that of STDMA. A full system simulation using GEM5 shows that the execution time of the proposed methods decreased by 6.5% in the four-chip stack and 17% in the eight-chip stack compared with that of STDMA.

I. Introduction

The conventional scaling of a transistor has finally started to reach its limits. 3D chip stacking is believed to be an alternative way to easily obtain a large number of transistors because of its cost [1].

There are two classifications for interconnecting between stacked chips, wired and wireless. Although wired interconnections (e.g. wire bonding, micro-bump bonding [2], and through-silicon-via (TSV) [3]) have been the mature techniques, the replacement, addition, and deletion of the stacked chips are not allowed once they are produced. On the other hand, with wireless interconnections (e.g. capacitive coupling [4] and inductive coupling [5]), the structure of the stacked chips can be easily changed because they are physically contact-less. While capacitive coupling is only for the face-to-face stacking of two chips, three or more chips can be stacked using inductive coupling interconnection. We focus on the inductive coupling Thru-Chip Interface (TCI) in our reseach.

Short distant vertical links, which can be used for a vertical bus, are an advantage of 3D stacking. While a number of studies on the vertical bus with TSV have been conducted [6] [7], only a few have forcused on a bus with TCI [8]. Since the footprint of the inductor is larger than that of the area for TSV, it is difficult to provide a lot of control wires. Since the precise synchronization of the clocks supplied to multiple chips in the stack is difficult, using a short time slot between multiple chips is also difficult. Thus, a simple static time division multiplexing access (STDMA) using a long time slot whose performance and scalability is severely limited has been used [9].

As a practical solution, we first propose an asynchronous TDMA bus (A-TDMA bus) using a collision detection mechanism. It is a type of STDMA bus without any synchronization clocks between the chips. Then, we propose a resonant synchronous TDMA bus (RS-TDMA bus) that uses a resonant synchronized clock for generating a time slot. They are compared with the conventional simple STDMA bus and dynamic TDMA (DTDMA) bus which is not practical for the TCI due to the number of inductors needed for the control signals. The evaluation results when assuming the use of a 3D microprocessor are also shown.

The rest of the paper is organized as follows. Section II introduces the TCI. Section III surveys applying conventional 3D bus architecture to the TCI bus. The A-TDMA bus and RS-TDMA bus are proposed in Sections IV and V. Then, Section VI evaluates the proposed methods, Section VII examines the cost of the inductor and implementation, and Section VIII concludes the paper.

II. Inductive Coupling Through Chip Interface

A. Inductive coupling channels

Inductive coupling TCI uses square coils implemented with common metal layers. As shown in Fig. 1, an inductive coupling channel is formed between two chips by stacking a transceiver coil on the receiver coil of a different chip. Two coils, one for the clock and the other for the data are

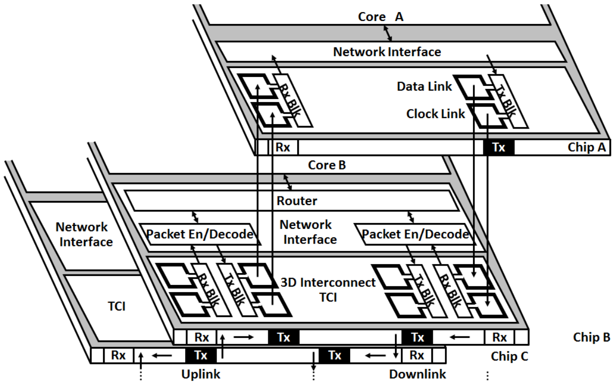

Figure 1. TCI with transmitter and receiver [11]

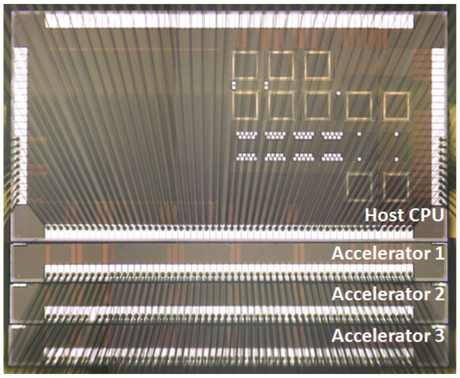

Figure 2. Chip stacking used in Cube-1 [11]

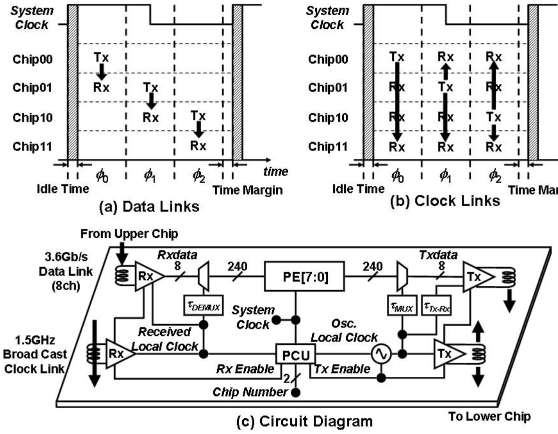

Figure 3. Bus used in MuCCRA-Cube [9]

usually provided for a channel. A high frequency clock (1 - 8 GHz) is generated using a ring oscillator, and the data are serially transferred in synchronization with the clock directly through the driver. The driver and inductor pair for sending data is called the TX channel, while the receiver and inductor pair is called the RX channel. Only a maximum of 8 Gbps of data can be transferred with a low energy dissipation (0.14 pJ / bit) and a low bit-error rate (BER$< 10^{-12}$) [10].

Data multicast can be used if a TX channel is placed at the same location as multiple RX channels on different chips. On the other hand, stacked multiple TX channels in the same location cannot simultaneously send data to avoid interference. Since a coil can be used for both the transmitter and receiver, the functionality of the TX and RX channels can be quickly switched, that is, a half-duplex bi-directional channel can be formed using a single coil.

Although TCI requires a certain amount of logic to form a link between two chips, it has the following benefits.

- A number of chips can be stacked if a physical environment is allowed.
- Since the chips can be tested before stacking, only known-good-dies can be connected.
- Since TCI is electrically contact-less, no electro-static-discharge (ESD) protection device is needed.
- Since the coil uses the common wire layers of a CMOS process, no extra process is needed. Although a coil has a large footprint, we can implement circuits inside the coil.

B. *Chip stacking and inter-chip networks*

Although a number of practical systems have been developed using TCI, most of them use a simple ring network. Fig. 2 shows the chip stacking used in Cube-1 [11]. The chips are shifted and stacked in order to place the receiver coil directly on the transceiver coil. The shifted space is also used to maintain the space for the wire bonding. Note that even in the TCI, several wires are needed for the power supply. In Cube-1, a ring-like packet switching network is formed just by stacking chips.

A statically time-multiplexed bus with TCI has been used in a dynamically reconfigurable processor called MuCCRA-Cube [9]. There can only be four stacked chips in this system, and the bottom chip does not send data. As shown in Fig. 3, each time slot $(\phi_0 - \phi_2)$ is statically assigned to each chip. In addition, a relatively low (50 MHz) system clock is distributed to all the chips to synchronize the time slot. A certain idle time and time margin are provided for compensating for a skew between the system clocks. Sixteen buses are provided for each processing element in order to support enough bandwidth for the poor bus throughput.

III. 3D Bus Architecture for TCI

A. *Static Time Division Multiple Access*

The bus used in MuCCRA-Cube is classified as a static TDMA (STDMA) bus in which each chip is statically assigned to a certain time-slot. Only the chip in the master time-slot can receive the bus grant and send data. Controlling the STDMA bus is relatively easy, since a complicated transmission control as well as control signals are unnecessary. In this method, the bus grant is evenly provided among all the stacked chips. However, since the allocation of the time-slot is always done statically, there will probably be an unnecessary waiting time for the bus grant. In particular, the efficiency of the bus utilization is degraded because of a long waiting time for the allocated time-slot when the number of stacked chips is large. Moreover, management of the time-slot requires synchronization clocks for each chip. A low frequency system clock (50MHz) is distributed in MuCCRA-Cube, and thus, it keeps to the given time margins so that the bus synchronization can be done. However, it is difficult to distribute a precisely synchronized clock to all the chips when we use a high frequency clock.

B. *Dynamic Time Division Multiple Access*

In dynamic TDMA (DTDMA) [7], each chip issues a request for bus mastership to a centralized arbiter when it is ready to send a packet. Then, the arbiter dynamically provides a certain time-slot to each chip depending on their requests. In this method, the unnecessary waiting time for bus mastership, which is a drawback of STDMA, can be avoided, since the time-slot is given to the chip on demand. Although the performance when using DTDMA is much higher than that when using STDMA, two problems arise when applying DTDMA to the TCI bus.

First, replacing, adding, and removing the chips in a stack are an important advantage of a 3D multiprocessor with the TCI. However, in the DTDMA, an arbiter must be placed on a stacked chip, and the centralized control will degrade the flexibility. Second, a number of point-to-point links for sending the request signals and receiving the time-slots between the stacked chips are necessary for implementing the arbiter with the TCI. Although the TCI is efficient when it is used for sending packets, it is inefficient for controlling the signals considering the size of a coil. In particular, the size of coil tends to be large when it is used for the bus, since the size depends on the distance between two communicating chips. Providing a number of large area coils only for sending a few bits of a control signal is a waste of the semiconductor area even if the logic can be implemented inside the coil. Like in STDMA, a synchronized clock must be distributed to all the chips, which will make the implementation difficult.

Here, we propose two practical methods that does not require centralized arbiters nor control signals.

At the moment, the available example of an allocation method for a bus grant in a 3D multiprocessor is STDMA [9] [12], which has a simple communication interface. However, as mentioned in Section III-A, STDMA has a poor bus utilization efficiency. Therefore, in this paper, we propose an asynchronous TDMA bus (A-TDMA bus) and a resonant synchronous TDMA bus (RS-TDMA bus) as feasible methods to decrease the latency of communications using the TCI bus.

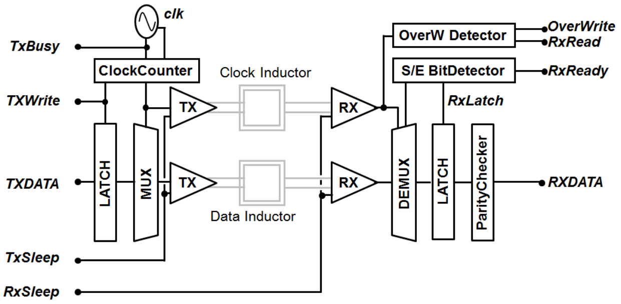

Figure 4. Intellectual property of bus architectures

IV. A-TDMA BUS

A. *Structure of A-TDMA bus*

Although most traditional 3D buses assume a synchronous clock without any skew, it is especially difficult to distribute it when the number of chips is large. Here, we propose an asynchronous TDMA bus (A-TDMA bus) that makes use of the flexibility of the TCI. In the proposed bus, we use a TCI whose transmitting (T_x) and receiving coils (R_x) are placed in a duplex-winding for both the data and the clock. A transceiver, a receiver, and the serializer/deserializer (SERDES) are provided when using such a data coil. For the clock coil, a clock generator, a transceiver, and a receiver are connected. A pair of coils containing the above mentioned modules form the intellectual property (IP) for easily building a bus with the TCI.

Fig. 4 shows a diagram of the IP for the bus architecture. The proposed bus controller is designed based on the IP. The important point of this IP is that the receiver can always monitor the bus and it is possible to know whether the bus is used or not. Since the transceiver and receiver can also work together, any collision on the bus can be detected by comparing the sending data with the receiving data.

B. *CSMA/CD in the A-TDMA bus*

We can introduce the conflict resolution methods used in LANs to the TCI using the bus sensing and colision detection mechanism of the bus IP. CSMA/CD [13] is the most widespread method used in a local area network (LAN). In this method, each chip receives all the data through the bus using its receiver. When a chip is ready to send a packet on the bus, it confirms that its receiver has not received a packet on the bus; that is, the bus is not used by others. Then, it tries to send the packet. Even with this operation, if two chips send packets exactly at the same time, the packets will come into conflict with each other. In this case, these chips cancel sending their packets during that cycle and compute the back-off time, which is the waiting time for sending and will never independently be the same values. Then, each chip waits for the computed back-off time before trying to resend to avoid the conflict again.

Like in the media for a LAN, each receiver can always receive the data on the TCI bus, and it can check whether or not data is being sent on the bus. If the data are not on the bus, all of the stacked chips have the right transmission. Then, each chip can send a packet using the bus. Otherwise, all of the chips have to wait to send until a tail flit in the packet is transferred, and a chip that wants to transfer data tries to send a packet on the bus.

In the way of mentioned above, the packets might conflict with each other because more than one chip might try to send packets at the same time. Here, the packet conflict in the TCI bus is detected as follows. Each of the stacked chips has a unique chip ID, and the ID is always inserted in the header flit of the packet when the packet is sent.

The sender chip can directly receive the packet that the sender itself has sent since duplex-winding has taken place between the transmitting and receiving coils. The sender chip checks the chip ID in the header flit of the packet. If the ID is different from the one that the sender itself has sent, it is then aware that other chips have sent packets at the same time.

C. Computing back-off time and resending packets

When a chip detects the collision of packets, it cancels sending its packet and then needs to resend it later. In the collision detection with the TCI bus, each chip is reminded how many times their sent packets experienced a conflict. However, the information related to other chips, such as which packet conflicted with its own and how many packets conflicted with each other at the same time, remains unknown. Therefore, each chip has to independently compute their back-off time.

In an Ethernet, a random exponential back-off ($CW = 2^n$) is used as the back-off algorithm using the computation of a contention window. In this algorithm, the contention window increases exponentially depending on n, the number of times packet has been resent. The contention window has an upper bound, that is, when n reaches a certain limitation, the value of the contention window no longer increases. We used the random exponential back-off in our A-TDMA bus.

V. RS-TDMA BUS

A. Resonant synchronous clock distribution

Although the inefficiency of the asynchronous bus can be solved by using a synchronous clock, distributing high speed synchronous clocks without a skew is difficult. The inductive coupling TCI can also be used for solving this problem by the resonant synchronous mechanism. Here, we use a coupled resonator with the inductors [14] in order to distribute a low-skew and low-jitter clock for the synchronization between stacked chips. As shown in Fig. 5, the inductive coupling TCI is used to distribute the clock vertically. We can lock the frequency and phase between the reference clock from outside and the clocks in each chip using the frequency-locking and phase-pulling (FL-PP) [15].

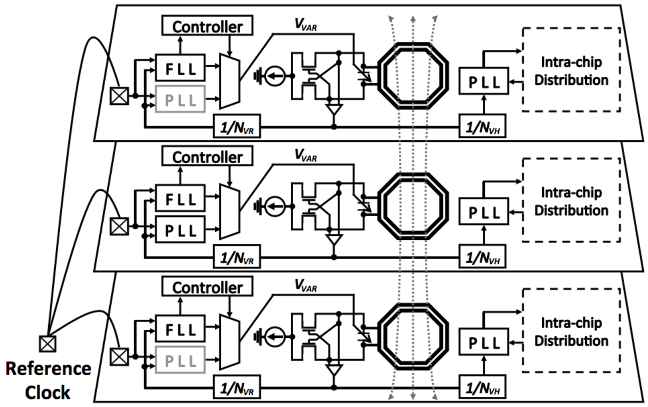

Figure 5. Resonant clock synchronization mechanism

Although this locking phenomenon was found about 40 years ago, it was recently applied to the TCI, and thus, a low-skew/low-jitter distribution was demonstrated[15].

B. Bus request and detecting conflict of request

Each chip can send a bus request before it sends a packet using the bus in order to avoid the wasted waiting time by using the resonant synchronization clock. The request can be sent in each cycle since the clock is synchronized among each stacked chip. When the request does not conflict, the chip can use the bus in the next cycle and send a packet on the bus. Whether or not the packet reaches the bus before a cycle can be recognized by using this routing method, and thus, the chip can send a bus request without using a redundant cycle.

If more than one chip tries to use the bus at the same time, there will be a conflict between them. This conflict can be detected by the analog circuits of the TCI like the A-TDMA bus. When requests do conflict, the states of the bus controllers of all the stacked chips switch to the resend mode. Detecting a conflict and changing the mode are done as follows.

The way to detect a conflict between bus requests differs in the chips sending the requests and the others.

- The chip sending a request receives a request signal that is different from the one it sent. Then, the chip can recognize that other chips have also sent requests at the same time. It then cancels its bus request in the next cycle, and switches its state to the resend mode.
- A chip that is not sending a request receives the request signals from other chips and waits to receive packets from the other chips in the next cycle. However, it does not receive a packet when requests conflicts with each

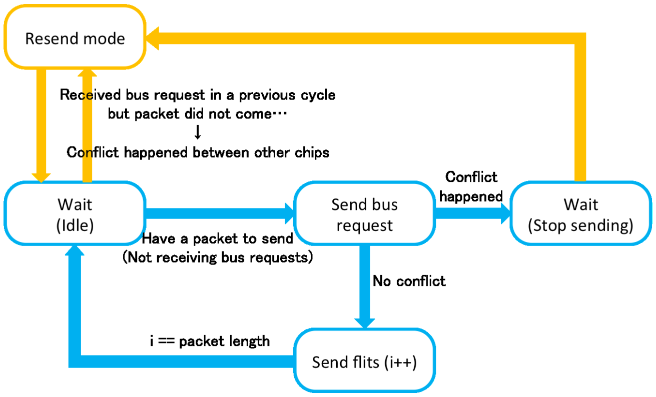

Figure 6. State transition diagram for RS-TDMA bus

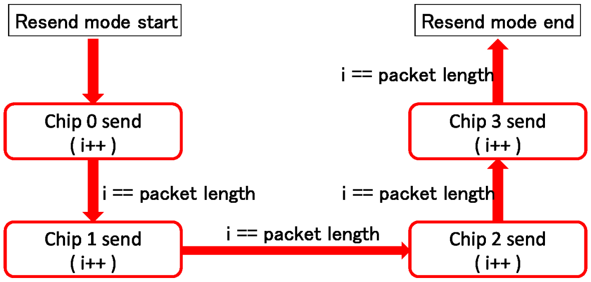

Figure 7. Resend mode

others. Thus, it can recognize that there are conflicting requests, and changes its state to the resend mode.

Here, we call a synchronous bus with the above mentioned control mechanism a RS-TDMA bus. The state transition of the bus controller is shown in Fig.6.

C. Resend mode

In the resend mode, each chip uses the bus in turn according to the previously defined order, as shown in Fig.7. When the chip with a request takes a grant, it sends a packet. When the chip without packet request waits for a cycle, the grant is given to the next chip. By doing this repeatedly, the resend mode ends after the grant is given to the last chip, and then, the state switches to the normal mode.

In the A-TDMA bus, a random back-off of the CSMA/CD is used for the resend method. This method causes a lot of conflict between the resent packets and the other packets when the communication load is high, and thereby, the number of resending packets increases. The latency due to the back-off time also increases since the value of the contention window rises. As a result, the communication performance degrades.

On the other hand, the packets never conflict when the packet is resent in the RS-TDMA bus owing to the resend mode. In addition, resending the packets using this method can be done with less overhead than using CSMA/CD, which makes the packets wait for a random time while resending.

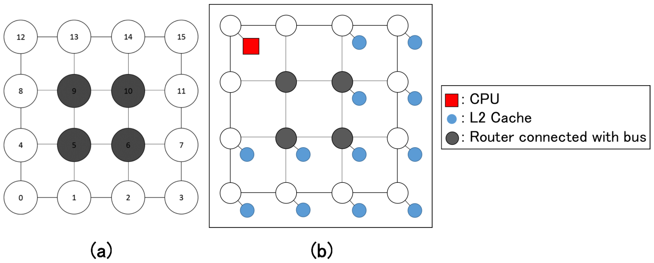

Figure 8. Chip structure: (a) Network performance and (b) Application execution

Table I
SIMULATION PARAMETER

Processor	x86_64
L1 instruction/data cache size	32kB
L1 cache latency	1 cycle
L2 cache size	256kB
L2 cache latency	6 cycles
Memory size	4GB
Router pipeline stage	3 cycle
Control packet size	1 flit
Data packet size	5 flits

VI. EVALUATION

A. Experimental environment

The proposed bus control methods are evaluated from two aspects in this section. The first is evaluating the network performance and the second is evaluating the application execution performance of the 3D multiprocessor with the proposed buses. We used GEM5 [16], a simulator for multicore processors.

Each target chip in the simulation has 4×4 mesh topology shown in Fig. 8. A chip stack using four or eight chips is the simulation target. Minimum hop routing is applied as the routing algorithm in the 3D multiprocessor. Six virtual channels are provided for each router to avoid deadlocks.

A simple STDMA was evaluated to compare it with the two proposed methods. The DTDMA, which is difficult to be adopted in the TCI, is also included for comparison as an ideal performance. The time-slot of STDMA was set at eight clock cycles and it was assumed that the arbitration in the DTDMA can be done in a clock cycle.

B. Network performance evaluation

In the network performance evaluation, uniform traffic and bit-complement traffic were used as the traffic patterns in the 3D multiprocessor. The results of the four-chip stack are shown in Figs.9 and 10, and eight-chip stack are in Figs.11 and 12. The horizontal and vertical axes show the injection rate and average latency respectively.

In the four-chip stack, the latency at the minimum load of two proposed methods was lowered by 29% compared with that of the STDMA under both traffic patterns. When the communication load is low, the wasted waiting time

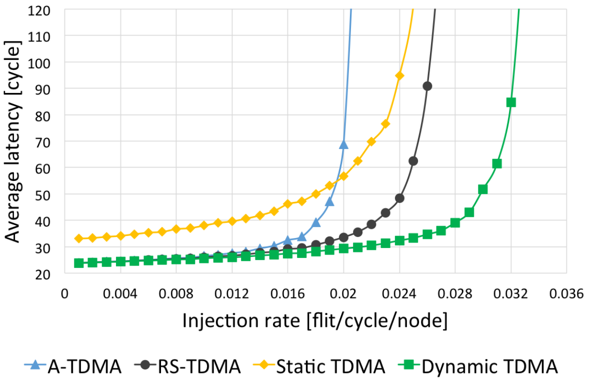

Figure 9. Average latency comparison (4-chip stack, Uniform traffic)

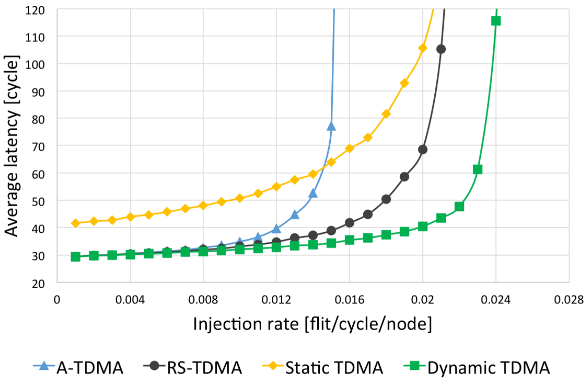

Figure 10. Average latency comparison (4-chip stack, Bit-complement traffic)

for using the bus tends to increase in the STDMA. On the other hand, the bus allocation in the two proposed methods is so efficient that waiting for bus usage rarely occurs. In the eight-chip stack, the minimum latency of the proposed methods were lowered by 50% compared with the STDMA. That is, the difference was larger than that of the four-chip stack. The proposed methods are only slightly affected by increasing the number of chips, while the waiting time for the bus is longer in the STDMA. These results show that the proposed methods efficiently allocate the bus and can be used for a chip-stack with up to eight chips.

In a comparison between the two proposed methods, the results show that the latency of RS-TDMA bus is lower than that of the A-TDMA bus when the injection rate increases to a certain extent. For example, the average latency of the A-TDMA bus suddenly increases while that of the RS-TDMA bus remains stable until about 0.02 when the injection rate is more than about 0.016 in Fig. 9. The way for resending packets in the RS-TDMA bus is more efficient because the packet never conflict in the resend mode. On the other hand, in the A-TDMA bus, conflicts tend to re-occur when there is a high communication load because of the random back-off.

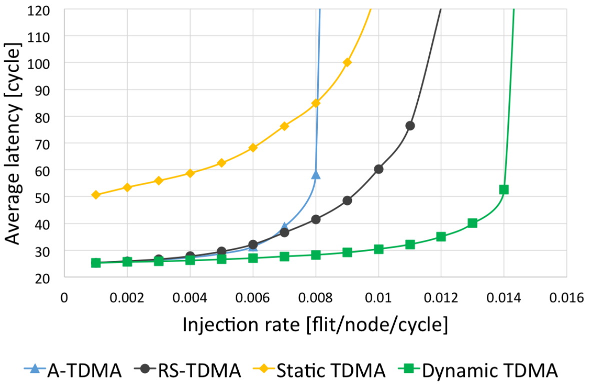

Figure 11. Average latency comparison (8-chip stack, Uniform traffic)

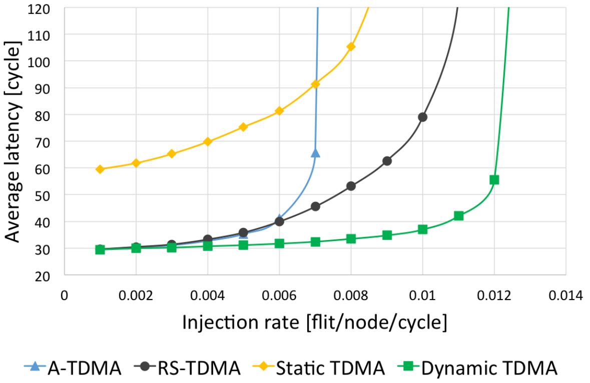

Figure 12. Average latency comparison (8-chip stack, Bit-complement traffic)

As a result, the network performance of the RS-TDMA bus is the closest to that of the DTDMA.

C. *Application execution performance evaluation*

Here, we evaluated the execution time of the NAS Parallel Benchmark (NPB) [17] using the GEM5 simulator. Seven benchmark programs were used for the full system simulation. Figs.13 and 14 show the execution times of each benchmark for the four-chip and eight-chip stacks.

According to these results, the execution performance of the eight-chip stack is especially improved, just like in the network simulation results, because the communication overhead between the chips has a greater affect in the STDMA than with the proposed methods.

VII. Cost and Implementation

The size of the inductor is relative to the distance between the transceiver and receiver. For safety, a side should be about twice the chip distance. Assuming an eight-chip stack with a 40 μm thick chip and 5 μm of glue, the distance is 355 μm, and thus, a 710μm-square coil is required. Of course, we must implement the digital logic inside the coil to

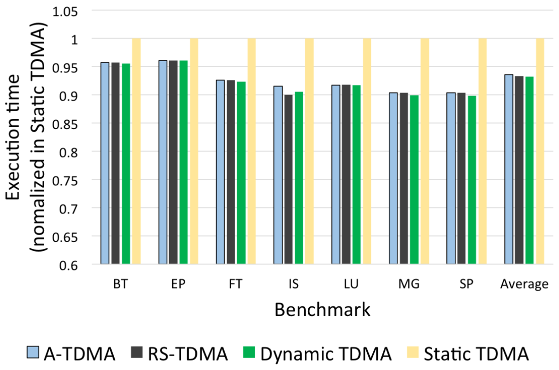

Figure 13. Application execution time (4-chip stack)

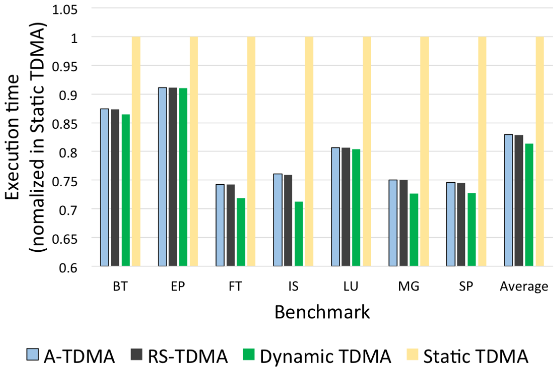

Figure 14. Application execution time (8-chip stack)

avoid wasting too much semiconductor area for the bus. In this case, the overall required area is for the clock generator, transceiver, receiver, and SERDES. Although it depends on the process technology, an area corresponding to hundreds of standard cells was required in Cube-1 [11]. Note that this fundamental cost for the TCI bus is even required for the STDMA bus.

Both the A-TDMA bus and RS-TDMA bus can be implemented using a standard router for the NoC, as discussed in a previous paper [8]. The controller for both buses requires from 2-3% of the hardware for the total router. The RS-TDMA bus requires a resonant synchronization mechanism that requires a large amount of hardware, which might not be worth the performance improvement of the RS-TDMA bus. However, this exactly synchronized clock can be used for various purposes. The RS-TDMA bus is advantageous in systems that require the resonant synchronization mechanism for other purposes.

Stacking chips with a bus is difficult if the wire bonding space must be maintained. We can use the stacking method proposed for MuCCRA-cube [9] in such cases. As shown in Fig. 15, each chip is stacked with a 180° rotation, and thus, the space for the bonding wires can be maintained.

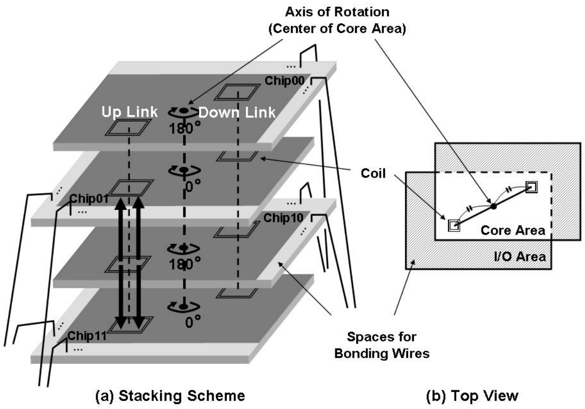

Figure 15. Bus used in MuCCRA-Cube [9]

VIII. Conclusion

We discussed our investigation of an efficient bus control mechanism for chip-stacks using the TCI, and proposed the A-TDMA bus with a CSMA/CD mechanism, and the RS-TDMA bus, which uses a resonate synchronized clock and a look-ahead mechanism.

The minimum-load latency of the two proposed methods decreased by 29% in the four-chip stack and by 50% in the eight-chip stack of STDMA by improving the efficiency for utilizing the bus. The communication performance of the RS-TDMA bus was closer to the DTDMA than that of the A-TDMA bus by avoiding conflicts of resent packets. The application execution time evaluation results showed that the execution time of the proposed methods decreased by 6.5% in the four-chip stack and by 17% in the eight-chip stack compared with that of the STDMA.

We are now developing an IP for the bus that can be used for both the A-TDMA bus and RS-TDMA bus based on the result of this study.

IX. Acknowledge

This work is partially supported by JSPS KAKENHI S grant number 25220002.

References

[1] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3D ICs: The Pros and Cons of Going Vertical," *IEEE Design and Test of Computers*, vol. 22, no. 6, pp. 498–510, Nov. 2005.

[2] K. Kumagai, C. Yang, S. Goto, T. Ikenaga, Y. Mabuchi, and K. Yoshida, "System-in-Silicon Architecture and its application to an H.264/AVC motion estimation fort 1080HDTV," in *Proceedings of the International Solid-State Circuits Conference (ISSCC'06)*, Feb. 2006, pp. 430–431.

[3] J. Burns, L. McIlrath, C. Keast, C. Lewis, A. Loomis, K. Warner, and P. Wyatt, “Three-Dimensional Integrated Circuits for Low-Power High-Bandwidth Systems on a Chip,” in *Proceedings of the International Solid-State Circuits Conference (ISSCC’01)*, Feb. 2001, pp. 268–269.

[4] K. Kanda, D. D. Antono, K. Ishida, H. Kawaguchi, T. Kuroda, and T. Sakurai, “1.27-Gbps/pin, 3mW/pin Wireless Superconnect (WSC) Interface Scheme,” in *Proceedings of the International Solid-State Circuits Conference (ISSCC’03)*, Feb. 2003, pp. 186–187.

[5] N. Miura, D. Mizoguchi, M. Inoue, K. Niitsu, Y. Nakagawa, M. Tago, M. Fukaishi, T. Sakurai, and T. Kuroda, “A 1Tb/s 3W Inductive-Coupling Transceiver for Inter-Chip Clock and Data Link,” in *Proceedings of the International Solid-State Circuits Conference (ISSCC’06)*, Feb. 2006, pp. 424–425.

[6] M.Daneshtalab, M.Ebrahimi, and J.Plosila, “HIBS - Novel inter-layer bus structure for stacked architectures,” in *Proceedings of 3D Integration Conference*, 2011, pp. 1–7.

[7] T. D. Richardson, C. Nicopoulos, D. Park, V. Narayanan, Y. Xie, C. Das, and V. Degalahal, “A Hybrid SoC Interconnect with Dynamic TDMA-Based Transaction-Less Buses and On-Chip Networks,” in *Proceedings of International Conference on VLSI Design (VLSID’06)*, Jan. 2006, pp. 657–664.

[8] T. Kagami, H. Matsutani, M. Koibuchi, and H. Amano, “Headfirst Sliding Routing: A Time-Based Routing Scheme for Bus-NoC Hybrid 3-D Architecture,” in *Proceedings of the International Symposium on Network on Chip (NoCS’13)*, Apr. 2013, pp. 29–36.

[9] S. Saito, Y. Kohama, Y. Sugimori, Y. Hasegawa, H. Matsutani, T. Sano, K. Kasuga, Y. Yoshida, K. Niitsu, N. Miura, T. Kuroda, and H. Amano, “MuCCRA-Cube: a 3D Dynamically Reconfigurable Processor with Inductive-Coupling Link,” in *Proceedings of the Field-Programmable Logic and Applications (FPL’09)*, Sep. 2009, pp. 6–11.

[10] N. Miura, H. Ishikuro, T. Sakurai, and T. Kuroda, “A 0.14pJ/b Inductive-Coupling Inter-Chip Data Transceiver with Digitally-Controlled Precise Pulse Shaping,” in *Proceedings of the International Solid-State Circuits Conference (ISSCC’07)*, Feb. 2007, pp. 358–359.

[11] N. Miura, Y. Koizumi, Y. Take, H. Matsutani, T. Kuroda, H. Amano, R. Sakamoto, M. Namiki, K. Usami, M. Kondo, and H. Nakamura, “A Scalable 3D Heterogeneous Multicore with an Inductive ThruChip Interface,” *Micro, IEEE*, vol. 33, no. 6, pp. 6–15, Nov 2013.

[12] Y. Take, H. Matsutani, D. Sasaki, M. Koibuch, T. Kuroda, and H. Amano, “3-D NoC with Inductive-Coupling Links for Building-Block SiPs,” *IEEE Transactions on Computers(TC)*, 2012.

[13] ANSI/IEEE std 802.3, “IEEE Standards for Local Area Networks: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications,” 1985.

[14] R. Adler, “A Study of Locking Phenomena in Oscillators,” *Proceedings of the IEEE*, vol. 61, no. 10, pp. 1380–1385, Oct 1973.

[15] Y. Take, N. Miura, H. Ishikuro, and T. Kuroda, “3D Clock Distribution Using Vertically/Horizontally-Coupled Resonators,” in *Proceedings of the International Solid-State Circuits Conference (ISSCC’13)*, Feb 2013, pp. 258–259.

[16] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, “The Gem5 Simulator,” *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.

[17] H. Jin, M. Frumkin, and J. Yan, “The OpenMP Implementation of NAS Parallel Benchmarks and Its Performane,” in *NAS Technical Report NAS-99-011*, Oct. 1999.