PPRAM-Link : A New High-Speed Communication Interface Standard for Merged-DRAM/Logic Systems-on-a-Chip Architecture

Hashimoto, Koji
Department of Computer Science and Communication Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Yamasaki, Masaya
Department of Computer Science and Communication Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Tomita, Hiroto
Department of Computer Science and Communication Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Metsugi, Katsuhiko
Department of Computer Science and Communication Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

他

出版情報：Proceedings of 3rd International Conference on SCI-based Technology and Research (SCI-Europe 2000), 2000-08
バージョン：
権利関係：

PPRAM-Link : A New High-Speed Communication Interface Standard for Merged-DRAM/Logic Systems-on-a-Chip Architecture

Koji Hashimoto, Masaya Yamasaki, Hiroto Tomita, Katsuhiko Metsugi, and Kazuaki Murakami

Abstract— **"PPRAM" (Parallel Processing Random Access Memory) is a novel architectural framework for ASSPs and integrates the following onto a single SOC, memory (DRAM or others), logic (MPU or application-specific logic), and network interface based on "PPRAM-Link" standard. The PPRAM-Link provides a high-bandwidth interface needed for communicating among two or more PPRAM nodes, which is composed of physical/logical layers and an API for the software development, these are standardized at PPRAM Consortium, based on SCI (IEEE Std 1596-1992). This paper describes the PPRAM-Link standard. As the physical layer standard, two Clock and Data Recovery (CDR) processing method are proposed. And then, the referential implementations, PPRAM-based system for *ab initio* MO calculations :"MOE", and PPRAM-Link I/F IP Core, are also described. Lastly, a prospect of coming step of PPRAM-Link is described.**

Keywords— **PPRAM-Link, PPRAM, Chip Interconnect, Merged-DRAM/Logic SOCs**

I. Introduction

A. Definition of PPRAM

Based on improvements in the degree of integration for transistors and the operation frequency of microprocessors, improvements in the performance of computer systems has been mainly attained in recent years as a result of following technique. Firstly, parallel processing for the instruction level of microprocessors, namely the pipeline processing and super-scalar processing are adopted. And an on-chip cache is embedded in microprocessors, furthermore, the cache architecture has improved. Secondly, the main memory capacity has increased.

However, limitations in the degree of instruction level parallelism have become obvious at the present time. Additionally, as the pipeline and super-scalar processing technique have become more advanced and complex, the design period and running cost, such as power consumption, have increased. Furthermore, the so-called "Von Neumann bottleneck", described below, has become apparent.

Firstly, the difference of performance improvement ratio between microprocessors and main memory (DRAM)s is increasing. An annual rate of latency time for DRAM row access is about 7%, and it is very little, while the operation frequency of microprocessors has been rising 22% per a year [1] [2] [3]. The other one is disclosure of limitations in memory access bandwidth. Various fast DRAMs with contracted column-cycle time and with extended data bus-width, have been thought out to ease preceding problems. But it is very difficult to go beyond 1GB/s in these effective memory access bandwidth, hence these new DRAMs (Direct Rambus, DDR SDRAM e.g.) will not be meet expectation in some applications.

To overcome the above-mentioned barriers against the performance improvement and to uplift the system performance, advancement of transistor density should be utilized for microprocessor/DRAM chips. We have examined a qualitative analysis from three viewpoints of the architectural complexity of processor, the number of processors embedded in a chip, and on-chip memory size in a microprocessor chip [6]. As a result of its investigation, we have introduced a conclusion that a system equipped with the following merits, namely the "PPRAM (Parallel Processing Random Access Memory)-Based System", is a powerful tool [4].

- Multiprocessing by using simple, multiple processors
- To mount high-speed processors/logic suitable for target applications
- To make the best use of higher bandwidth of on-chip memory
- To reduce power consumption by using embedded memory

We define that "PPRAM" is an architectural framework for ASSPs (Application-Specific Standard Products), in other word, an architectural concept that builds up computers or electronic equipment systems of any size, any function, or any performance according to the request of applications. This is accomplished by way of parallel/distributed interconnections of one or more merged-memory/logic SOC(Systems-on-a-Chip)s which are basic configuration elements, by using a standard interface for inter-chip interconnections. Figure 1 shows three technology backgrounds and significant keywords that PPRAM stand on.

In addition, "PPRAM node" is the basic configuration element of the PPRAM-based system, and it is composed of the three following elements. Figure 2 shows a concept of internal PPRAM node.

1. Memory of zero or more bits, which is composed of single memory units or a combination of memories such as DRAM, SRAM, and Flash EEPROM.
2. A homogeneous or heterogeneous multiprocessor, which is composed of zero or more logic units such as general-purpose MPU, DSP, and processors for specific applications.
3. One or more standard communication interfaces, which is called "PPRAM-Link", and it enable high-speed commu-

K. Hashimoto, K. Metsugi and K. Murakami are with the Department of Computer Science and Communication Engineering, Graduate School of Information Sceience and Electrical Engineering, Kyushu University, Kasuga-shi, 816-8580 Japan.(E-mail: koji@c.csce.kyushu-u.ac.jp ; metsugi@c.csce.kyushu-u.ac.jp ; murakami@c.csce.kyushu-u.ac.jp)

M. Yamasaki was with the Kyushu University, Kasuga-shi, 816-8580 Japan. He is now with the Computer on Silicon Development Center, Toshiba Corporation, Oume-shi, 198-8710 Japan (E-mail: masaya.yamasaki@toshiba.co.jp).

H. Tomita was with the Kyushu University, Kasuga-shi, 816-8580 Japan. He is now with the Corporate Semiconductor Development division Matsusita Electric Industrial Co.,Ltd., Fukuoka-shi, 814-0001 Japan (E-mail: tomita@qmc.csdd.mei.co.jp).

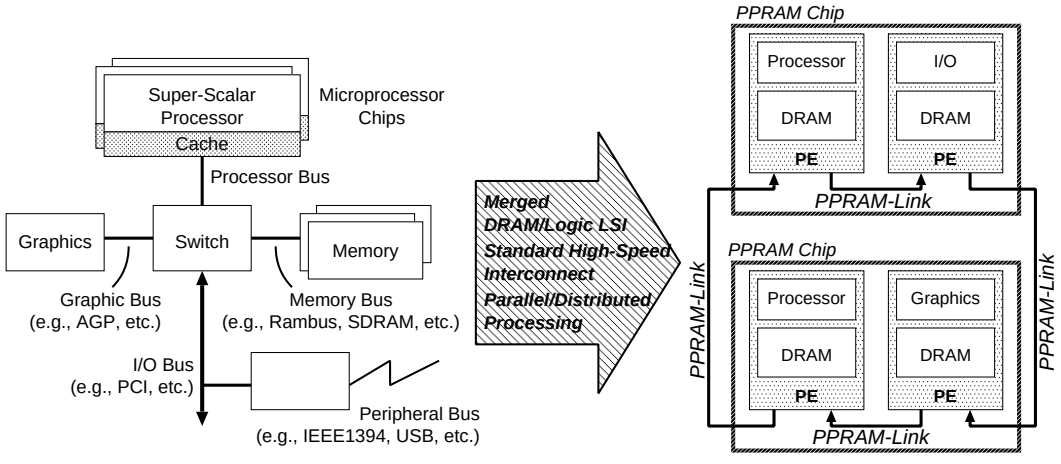

Fig. 3. PPRAM paradigm shift

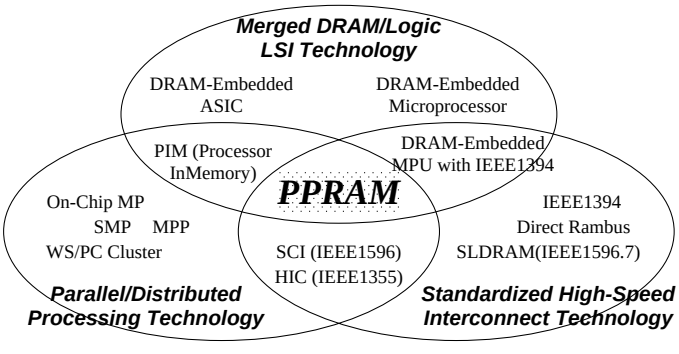

Fig. 1. Technologies that PPRAM stand on

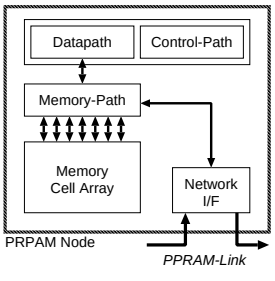

Fig. 2. Concept design of PPRAM node

nication in a PPRAM chip and between PPRAM chips.

Figure 3 shows an example of paradigm shift from the present computer system to the PPRAM-based System. Every PPRAM chip includes one or more PPRAM nodes, and two or more PPRAM nodes or chips are interconnected by PPRAM-Link. Consequently, the PPRAM-Based system means mainly a system which is composed of one or more PPRAM chips. Many different implementations of PPRAM chips and PPRAM-based system can be realized, but designers of those don't have to touch the communication protocols/interface. They have only to do decisions arbitrarily of the internal architecture according to target applications, concretely, size/type/bandwidth of memory and number/function/performance of processor. On the other hand, system designers have only to select desirable type/number of PPRAM chips according to the target application. However, inter-connectability, usability and portability between all the PPRAM chip vender should be assured to enable and promote above-mentioned technique for the LSI/system design. This is why the communication interface/protocol should be standardized.

As mentioned above, "PPRAM-Link" is the interface standard for typical high-speed communication between LSIs, in order to interconnect SOCs, namely PPRAM chips. PPRAM chips are connected to each other by a one to one single way serial or parallel link, targets for the near future are, in specific values, 1Gbit/sec or more for a serial link, and 1Gbyte/second or more for a parallel link.

The PPRAM-Link standard consists of a logical layer, a physical layer, and an Application Programming Interface (API) for the software development, as shown in Figure 4. The physical layer compensates undistorted/smooth transmissions, and it bears the roles that passes electrical/physical signals to the Logical Layer of its own node as symbols which is 16 bit-width. On the other hand, a set of communication protocols is provided at the logical layer. It possesses a transmission layer, transaction layer, and initialization/error processing protocol. The transmission Layer bears the roles of the flow-control to guarantee the right packet transmission, and at the transaction layer, a transaction instructed by an application layer, is implemented by packet exchanges.

The API standard provides a common interface for the development of low-level software like device drivers, DSM(Distributed Shared Memory) interfaces and runtime libraries. It aims to assure the inter-usability and portability on the PPRAM-based system, while it doesn't define specific operating systems or instruction-set of processors which is embedded in PPRAM chips.

Since 1997, PPRAM Consortium[11] has been making a continuous effort to develop open standard specifications for inter-chip interconnection, or PPRAM-Link. Foundation of the PPRAM-Link logical layer is mainly Scalable Coherent Interface (SCI, IEEE standard 1596-1992), and in addition, HIC (IEEE1355) and IEEE1394 are also refered partly.

We have been played an important role in the above-

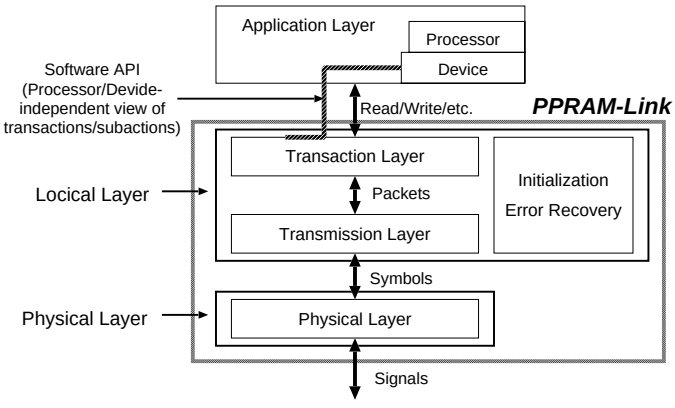

Fig. 4. PPRAM-Link protocol stack

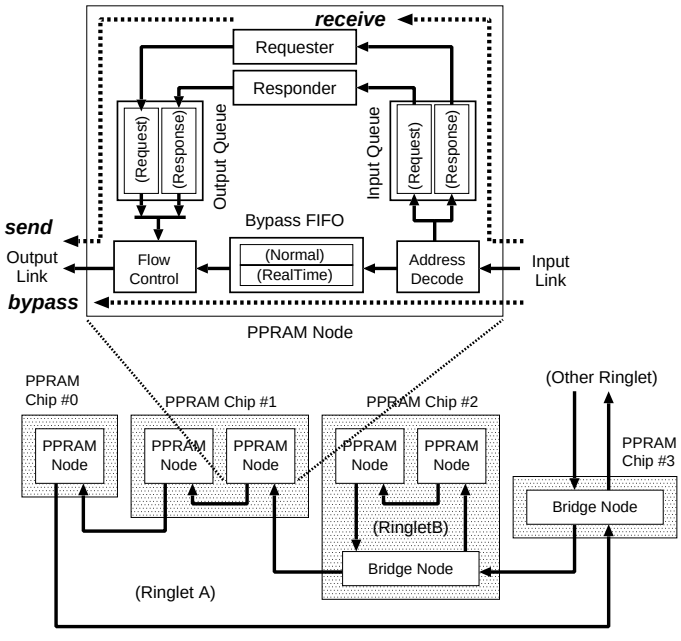

Fig. 5. Illustration of PPRAM nodes/chips interconnection (Upper part: Functional definement of a PPRAM node)

mentioned work, and the PPRAM-Link Standard Version 1.0 have laid down. In this paper, we propose the PPRAM-Link standard which consists of logical/physical layer and API interface, and then, referential design of PPRAM-Link physical layer.

Besides, we also propose a high-performance and special-purpose parallel machine for *ab initio* Molecular Orbital calculations, called MOE, and PPRAM-Link I/F IP core "PLIF Core" for multi-purposes.

Lastly, we mention a task to be discuss in the near future, and then, a new generation of the PPRAM standard.

II. PPRAM-Link standard

One PPRAM node is composed of zero or more bits of local memory, zero or more processor/logic units, and one or more network interfaces. The PPRAM node model is based on SCI model, so it is provided one or more output packet-queues and input packet-queues.

One PPRAM chip have one or more PPRAM nodes. Figure 5 shows a illustration of the PPRAM node and interconnections of PPRAM chips (part of a whole PPRAM-based system).

A. Requirement to PPRAM-Link

PPRAM-based system is parallel/distributed processing system that its memory area is divided to numbers of piece, or PPRAM nodes. And it try to gain both high adaptiveness and plasticity. So, the node interconnection standard for PPRAM-based system, named PPRAM-Link, requires to have scalable network functionality like existent networks used in existent high performance computer systems. But keeping the PPRAM-Link interface circuit in low-cost is indispensable, since there are physical constraints caused by that PPRAM-Link must be inter-chip communication network. On the other hand, existent standards for inter-chip bus connections, such as PCI and several fast DRAM interface (RambusDRAM,DDR SDRAM etc.) will not give good scalable network functionality. Therefore, we have found that our PPRAM approaches are get along with features of SCI standard, we are trying to construct PPRAM-Link with SCI as starting point.

As mentioned above, PPRAM nodes are connected to each other by a one to one single way serial or parallel link. The physical layer bears the roles that passes electrical/physical signals to the logical layer from one node to one node as continuous flow of symbols. And the logical layer doesn't matter whether the signal is transmitted in serial/parallel as long as the sufficient bandwidth can be derived.

Next, independently of the physical layer, the logical layer specification is composed of following protocols:

- Transaction protocol
- Flow control protocol
  (It contains Bandwidth allocation, Queue allocation and Real-time transmission.)
- Error recovery protocol
- Initialization protocol

PPRAM-based system is basically a NCC-NUMA (Non Cache Coherent, Non Uniform memory Access) architecture, so the logical layer itself doesn't have to contain cache coherent protocols that SCI supports. But it may be possible to supply coherent shared-memory spaces for application programs.

The transaction layer includes transaction protocol, and the transmission layer includes flow control protocol. Error detection and initialization protocols are lies across the whole logical layer. Figure 4 shows a illustration of PPRAM-Link protocol stack.

B. Logical layer

Protocol specifications of PPRAM-Link logical layer are basically as same as SCI, and it is numbered `1.0`. In this subsection, we summarize differences of logical layer protocols between PPRAM-Link and SCI.

As mentioned above,Transaction types related to a cache coherent protocol are not provided in PPRAM-Link. Consequently, the following transactions are excluded:

- `mread00, 64` : Coherent memory control/read
- `mwrite16, 64` : Coherent memory write

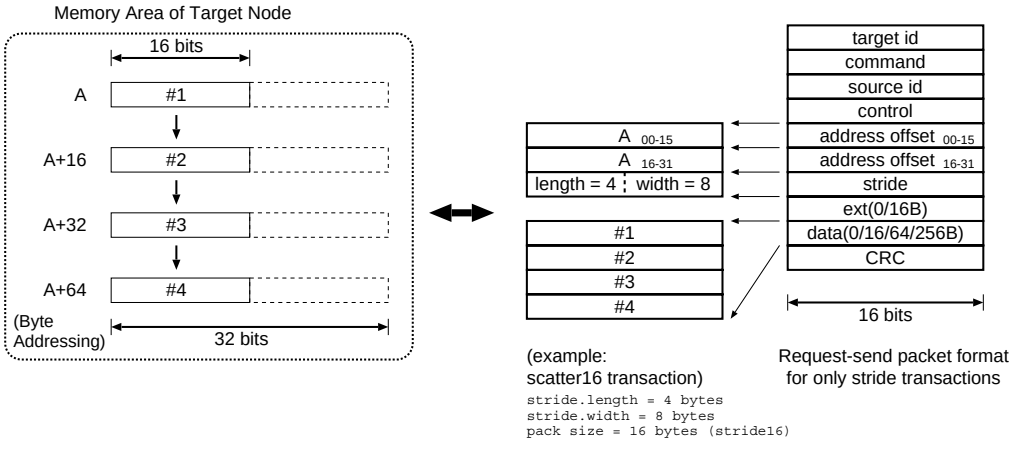

Fig. 6. Data transmission using PPRAM-Link stride transactions

- `cread00, 64` : Cache-to-cache control/read

On the other hand, new transactions for stride transferring are added. They are itemized as follows.

- `gather16,64,256` : Non-coherent memory read with stride access/transmission
- `scatter16,64,256` : Non-coherent memory write with stride access/transmission
- `active-message16,64,256` : Message transmission from PPRAM nodes to PPRAM nodes
- `suspend` : Suspend a target node, for power-saving etc.
- `resume` : Resume a target node, that has been suspended before

These transaction sets will be useful to transfer matrix rows/columns efficiently. Figure 6 shows an example of data transmission with scatter16 transactions, and the request-send packet format.

There are more than a few influences of the packet format specification upon circuit size of PPRAM-Link interface. Thus, we have modified the packet header format on the basis of the pilot circuit design. But there is no practical modification on the symbol fields concerned with flow-control and error-recovery protocols.

An unique identifier (UID) value is embedded in all of PPRAM nodes when they are manufactured, as same as SCI standard. PPRAM-based system is be partly based on Control and Status Registers (CSR) architecture (IEEE standard 1212-1991), so CSR registers are to be embedded in each PPRAM node.

We define a ringlet as the basic topology, and it is a vertex of the PPRAM-Link network, or PPRAM-based system. PPRAM nodes also feature their own node ID values of 16bits length that is assigned at the ringlet initialization, which is simplified. It is possible to connect a maximum number of 65535 nodes in a PPRAM-based system by applying an enhanced method of assigning node ID to the multiple-ringlet system.

The initialization protocol also needs to be simplified, because the sequential processing flow is mainly substantialized in a hardware logic. The digest is shown as follows.

- The number of nodes is less than 16 in a PPRAM-Link ringlet
- When address initialization phase, basic topology of ringlet interconnection is hierarchical tree, but it is possible to interconnect PPRAM nodes in non-hierarchical multi-ringlet

Besides, an address bit-width allocated to each PPRAM node is more than 32bits, and less then 96bits, which can be freely used by the upper application layer of PPRAM-Link communications.

C. Physical layer

C.1 Policy

With speed-up of microprocessors and increase of the data processing amount in a recent years, fast data transmissions is also required in a field of chip interconnection interfaces. But frequency of conventional interface technologies, such as Low-Voltage TTL, doesn't beyond 100 MHz. And it is caused mainly by decrease of timing budgets, distortion of signal waveform, and increase of power consumption.

In order to realize the high-frequency data transmission, it is indispensable to repress aforementioned three problems. For example, latest memory interface standards such as Direct Rambus and DDR SDRAM, or IEEE1394 (for inter-module connection), have been contend with aforementioned problems, by using their own original techniques. These techniques can be digested as follows.

1. Secure timing budgets and prevent distortion of signal waveform : Timing compensation and recovery by using a phase-adjuster such as DLL(Delay-Locked Loop), and strict rules of signal lines on circuit boards
2. Lower power consumption : Adoption of low-swing signal interface technologies

As for PPRAM-based SOCs that merged memory and logic, the number of times to access off-chip memory can be reduced. But in order not to block the advancement of system performance that results from the inter-chip connection interface, and to realize an efficient parallel/distributed processing that can surpass systems of conventional architecture (memory and MPU are separate chips), target of the communication performance are 1 GByte or more per second for a parallel link. Therefore, PPRAM-Link physical layer standard pro-

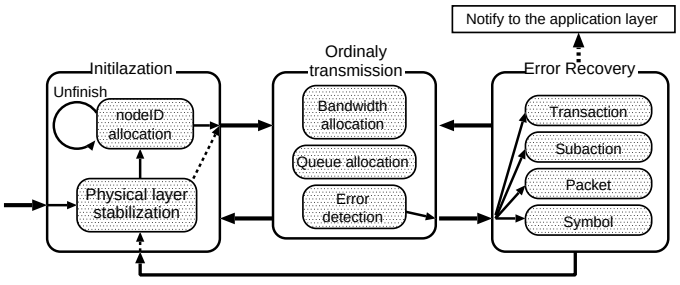

Fig. 7. Phase transition on PPRAM-Link logical/physical layer

vides under-mentioned two methods/specifications as prescriptions for high-speed inter-chip communication.

1. **Clock and Data Recovery (CDR) processing method** : A sequential processing to secure timing budgets is defined. Mainly, it proposes a phase-adjustment technique inserting DLLs in the data line.
2. **I/O interface specification** : At present time, existent I/O interface techniques such as Low Voltage Differential Signal (LVDS), Gunning Transceiver Logic (GTL), or Stub Series Terminated Logic (SSTL) are suitable for the present PPRAM-Link's goal.

This physical layer standard is numbered "1.0" . Aftertime, new I/O interface specifications will be needed. For example, Multi Chip Module (MCM) is often applied to the construction of SOCs, such as processing module that contain MPU, cache or I/O chips. At the present time, MCM method may be a good choice to develop SOCs, or PPRAM-based system, because it have high scalability and easiness to fabricate MCMs. Therefore we are now planing a new PPRAM-Link standard for MCM or other sub-chip module way, that will be renewal mainly of the physical layer and part of the logical layer.

C.2 Subsumption

Signal communication between PPRAM chips takes the following formation in conformity to the logical layer.

- Signal line-width : a parallel link with 17 bits (= 16 bits of a symbol + one flag bit)
- Connection : One to one unidirectional communication between two PPRAM chips

For the present, we assume a circuit board that multiple PPRAM chips are placed. Thus, communication between PPRAM nodes which are inside in a PPRAM chip is beyond of our subject.

At the physical layer, it is required that the stabilization phase starts after the PPRAM-Link ringlet have initialized/reseted, and the phase must become stable before actions of the logical layer starts.

Figure 7 shows a digest of phase transition on overall PPRAM-Link layers.

C.3 Clock and data recovery processing

According as purposes that PPRAM-based system has, several types of PPRAM-Link physical layer may be possible, and it can be classify into two topics, "How to feed a clock signal to PPRAM chips" and "Send data with a reference clock signal or not". Therefore, we provide two methods that is suitable for Clock and Data Recovery (CDR) processing, as follows. And they are shown in Figure 8:

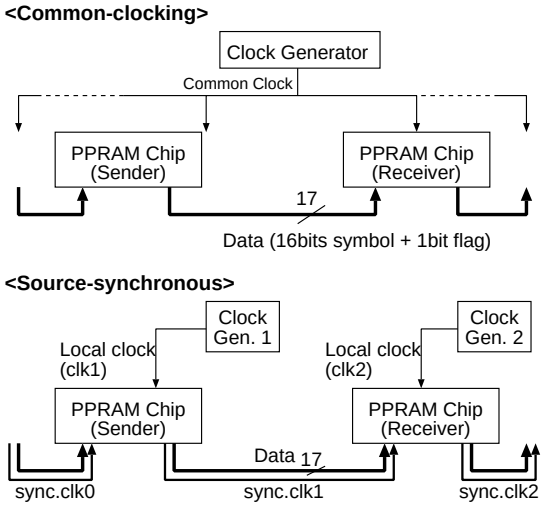

Fig. 8. Two ways to process CDR

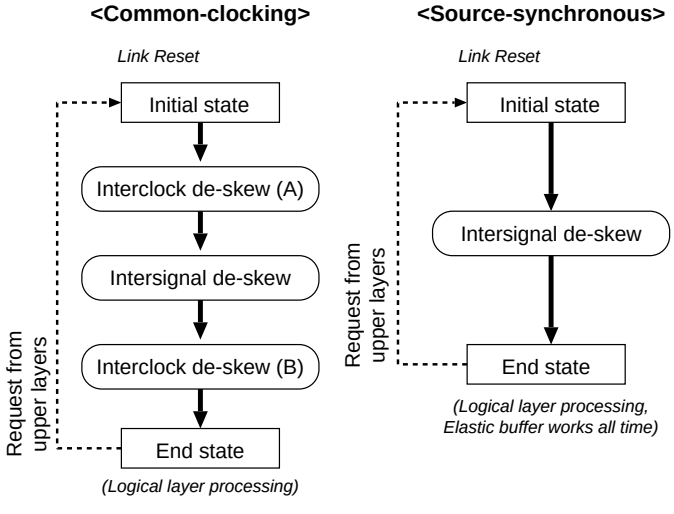

Fig. 9. State transition of two CDR methods

1. **Common-clocking method** : All LSIs that join in PPRAM-Link communication share a solitary clock source, so clock frequency of all sender/receiver nodes may be same. Hence, a receiver node can fetch data by its own clock, if the timing budgets is well secured by applying a inter-clock de-skew processing. Moreover, an inter-signal synchronism of 17 bit-width parallel data should be secured by a inter-signal de-skew processing. For above de-skew processing, DLL circuits are put to use.
2. **Source-synchronous method** : LSIs that join in PPRAM-Link communication have some clock sources, so a small jitter between clocks of sender/receiver node may be exist. To absorb the jitter, a elastic buffer is provided in the receiver node. And then, equally to the common-clocking method, the inter-signal de-skew processing is also needed.

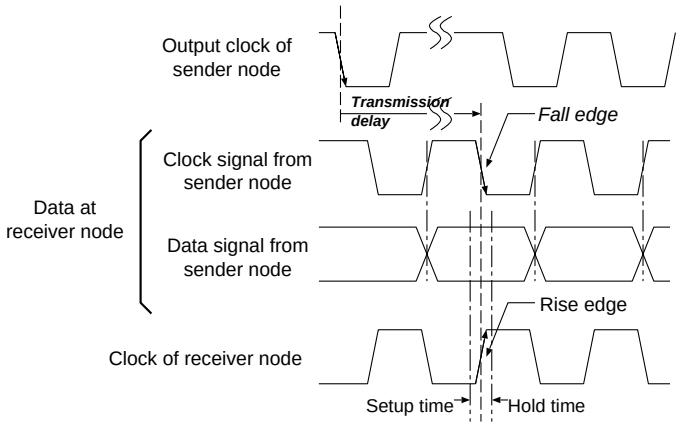

Fig. 10. Phase-adjustment to secure a timing budget

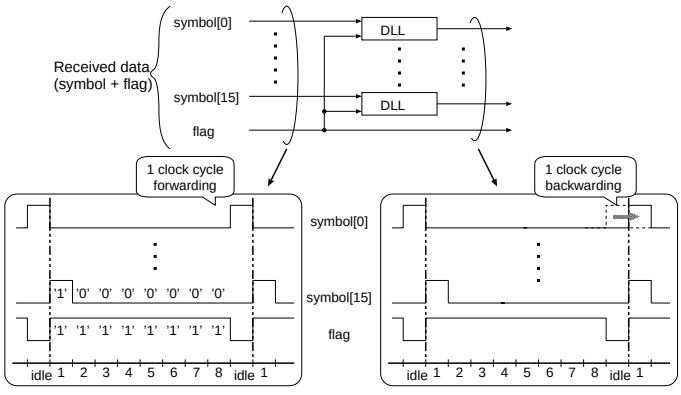

Fig. 11. Inter-Signal-de-skewing of 17bit-width data

Each method are composed of some internal states, as shown in Figure 9. State transitions in above-mentioned two CDR methods are controlled by a timer that embedded in each physical layer logic.

In the common-clocking method, there are three state mainly as follows.

1. Inter-Clock de-skew (A) : In order to secure timing budgets, delay time of each DLL inserted in own data bits, is adjusted to interlock a flip-flop setup/hold time with a valid-time of receiving data bit. For this phase-adjustment, the sender node sends clock signals in a 17bit-width data line. The receiver node adjusts each DLL to be own clock signal in anti-phase with a receiving data bit. Figure 10 illustrates the phase-adjustment to secure a timing budget, at the common-clocking method.
2. Inter-Signal de-skew : The above process enables the receiver node to fetch data in no-error. However, it is possible that each data bit is fetched by a different clock timing. So DLLs are needed to eliminate each inter-signal skew of 17 bits, and they are adjusted by the unit of one clock-cycle time. For the adjustment of above DLLs, a sync packet is used as same as the SCI. Figure 11 illustrates the inter-signal de-skewing at the common-clocking method.
3. Inter-Clock de-skew (B) : The inter-signal de-skew processing is more imprecise than the inter-clock de-skew processing, and phase-dispersion may be occurred. So this inter-clock de-skew processing is needed once again.

The common-clocking method don't need a elastic buffer, so the end state dose only keep a set point of DLLs.

Next, in case of the source-synchronous method, a reference clock is send with data, so the receiver node have only to fetch data by a inverted signal of the reference clock. But there is a little inter-signal skew of mutual data bits that should be eliminated, and DLLs are needed for it. Therefore, the source-synchronous method have only a state to process it. At the state, a sync packet is used for the adjustment of DLLs. And a elastic buffer works constantly to absorb a jitter between sender/receiver nodes' clocks.

III. Referential circuit design of PPRAM-Link physical layer

A. Examination of DLL architecture

We had examination and designing of PPRAM-Link physical/logical layer circuits that stand on above-mentioned specifications, to satisfy functional requirements. In this section, a referential design of physical layer circuit is described.

Each of two CDR methods has phase-adjustment functions by using DLLs. But each DLL requires multiple delay times and particle delay sizes. Therefore, we divide DLLs roughly into a "Fine DLL" for fine phase-adjustment and a "Coarse DLL" for coarse one.

Fine DLLs are used in the inter-clock de-skewing of the common-clocking method, and inter-signal de-skewing of the source-synchronous method. And then, Coarse DLLs are used in the inter-signal de-skewing of the common-clocking method.

As for the architecture of DLLs, the open-looped type and close-looped type can be produced. It is generally that former DLLs is capable of locking in a short time, but latter DLLs is capable of deriving extraordinary precision delay times [5]. This referential circuit design precedes highly the precision and easiness to design DLLs, so we adopt a digitally close-looped type with registers, for both of Fine and Coarse DLLs.

Figure 12 illustrates a block diagram that uses the common-clocking method. In that case, PPRAM chips contain 33 DLLs in the receiver, transmitter that select data for the CDR processing or ordinary data, and a logical block that control the whole physical layer. The transmitter is a simple selector that selects data from parallel clock signals for aforementioned inter-signal de-skew processing, and ordinary data from the logical layer. Figure 13 illustrates a block diagram that uses the source-synchronous method. It contains 17 Fine DLLs and one elastic buffer, but it doesn't contain a particular transmitter block.

All Fine/Coarse DLL is designed by a custom layout, and other logic circuits are designed by HDL auto-synthesis. The LSI design rule and preconditions are described as follows.

- 0.35 μm CMOS technology, with 3 metal and 1 poly-silicon layers
- Power voltage = 3.3 V, and internal/external frequency = 50 MHz
- I/O Interface : LVTTL

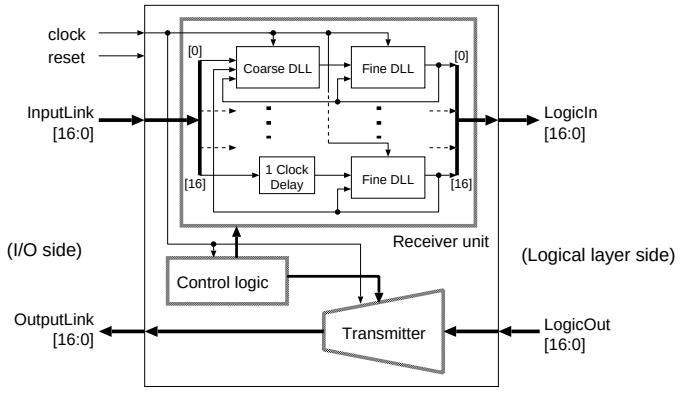

Fig. 12. Block diagram of the common-clocking method

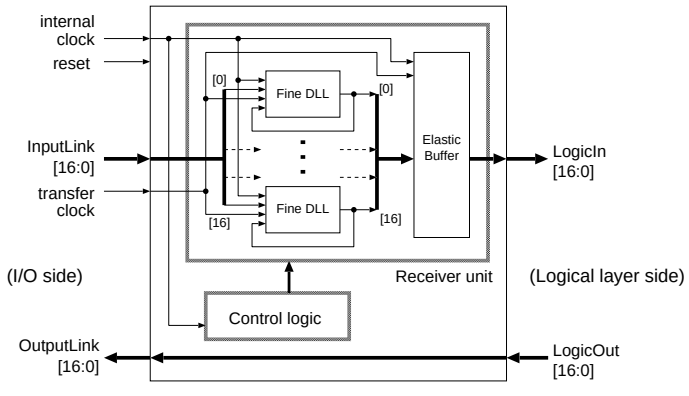

Fig. 13. Block diagram of the source-synchronous method

- Supposed signal delay on circuit board = 0.6 $\sim$ 1.0 ns /10 cm

There are two ways to make a desirable delay time, The one is use of weighted delay elements (combination of $\Delta t_d, 2\Delta t_d, 4\Delta t_d, ...$), and the other is use of same delay elements (a liner connection of Δt_d s). In this paper, the latter way is taken to simplify the circuit design.

All Fine DLLs are required to have phase-adjustment capacity at least half of clock-cycle ($180°$) at one side, that is, more than one clock-cycle ($360°$) totally. The phase-adjustment grain: Δt_d of the Fine DLL is determined arbitrarily by the accuracy of own phase-comparator and the allowed phase-drift. But in fact, a dispersion : $\Delta t_{d_{best}} \leq \Delta t_d \leq \Delta t_{d_{worst}}$ may be assumed from "*worst*" to "*best*", by dispersions of chips' process-designing and working condition. Therefore, dispersions should be taken account of when Fine DLLs are designed.

As the common-clocking method has two inter-signal deskew processings its Fine DLL must have phase-adjustment capability of two clock-cycle delay time, while Fine DLLs for the source-synchronous method should have of one clock-cycle delay time.

On the contrary, Coarse DLLs have only to be set the delay time at selection from -1 / 0 / +1 clock-cycle. So its structure may be easy.

Both of Fine/Coarse DLL are composed of a delay-line, a phase-comparator, and a N-bits counter. The delay-line is a linear connection of N delay-element. The delay-element is a

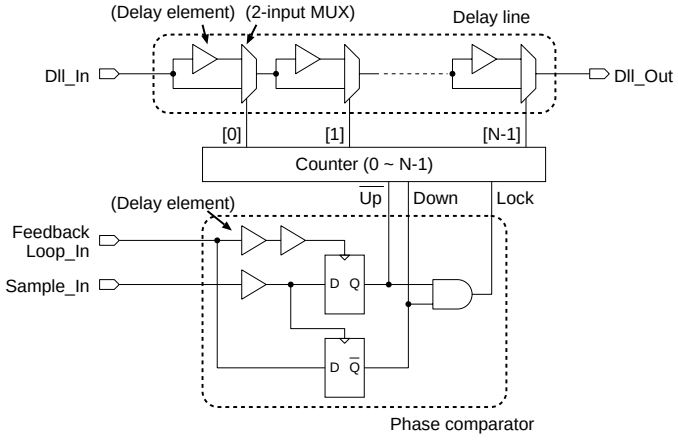

Fig. 14. Foundamental architecture of Fine DLL

combination of a delay-line and a two-input multiplexer. Figure 14 illustrates a fundamental circuit architecture of Fine DLL. In case of the common-clocking method, a clock signal of own node is used for its data sampling signal (`Sample_In`), and in case of the source-synchronous method, a flag bit signal sent with a symbol is used for its data sampling signal.

At a referential design outlined in this paper, all elements of delay-line are made from series of CMOS inverters, because it is important to acquire the stable and liner character that can be gained when delay-elements are connected in series.

Each Fine DLL has two types of delay-elements with different delay time, Δt_d for the delay-line and Δt_{comp} for the phase-comparator. At this referential design, the same delay-element is adopted for all delay-elements of delay time = 1 ns. And then, we have designed the Coarse DLL with delay time : Δt_d = one clock-cycle = 20 ns. Architecture of the Coarse DLL is basically as same as the Fine DLL. However, its phase-comparator, counter and delay-elements can be simplified.

B. Circuit estimation

Table I shows performance values of practical Fine/Coarse DLLs, as results of post-layout simulation using HSPICE. And Figure 15 shows changes of rising/falling delay time of Fine DLL, when varying the number of delay-elements. All of differences between the rising time and the falling time are less than 100 ps, hence it is obvious that the delay-line produce a good liner increasing of delay time.

To estimate and verify scales of physical layer circuit with common-clocking method and source-synchronous method, we have designed two referential circuits using above-mentioned Fine/Coarse DLLs.

At the common-clocking method, the physical layer circuit is composed of 3394 cells, and 92% of the whole circuit is used for the receiver. At the source-synchronous method, it is composed of 2616 cells, and 66% of the whole circuit is the receiver. The standard cell library contains a delay-element for Fine DLL, a series of Fine DLL's delay-elements is used as a delay-line for Coarse DLL.

Hence, it is important that how downsize these receiver logics, in case that the physical layer is applied to actual PPRAM

TABLE I
PERFORMANCE OF FINE/COARSE DLL

	Common-clocking		Source-synchronous
	Fine DLL	Coarse DLL	Fine DLL
Typical delay time : Δt_d [ns]	1	20	1
Best delay time : $\Delta t_{d_{best}}$ [ns]	0.63	10.3	0.63
Worst delay time : $\Delta t_{d_{worst}}$ [ns]	1.67	28.5	1.67
Maximum number of times to change delay time *	16	2	16
Number of delay-elements	66	3	33
Realizable maximum delay time [ns]	110.22	59.2	55.11

* : The value is determined automatically by the number of delay-elements in the DLL, and a maximun permissible time to lock the DLL.

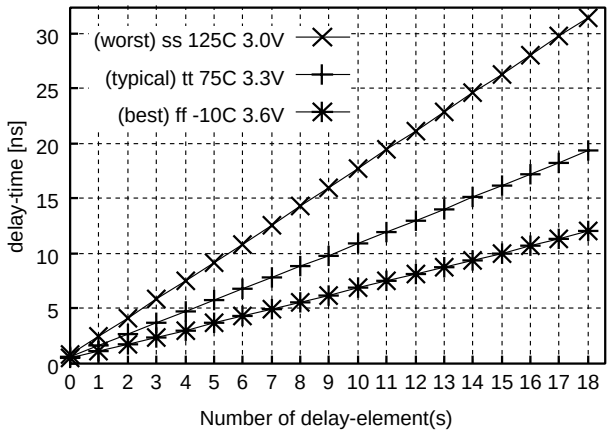

Fig. 15. Changes of rising delay time of Fine DLL

chips. But current logic circuits are very simple, it may be easy fundamentally to reduce the number of delay-elements in all DLLs.

IV. PPRAM-MOE

As the example of PPRAM-based system that we have developed, the MOE system and its processing node chip: "PPRAM-MOE" are described as follows.

A. Outline of MOE

Ab initio molecular orbital (MO) method presents important information on electronic states of molecules. But, large-scale *ab initio* MO calculations have at least two serious problems: *long computation time* and *poor numerical accuracy*. Because the computational complexity is $O(N^4)$ where N is the basis size of molecular systems, and most of time is spent to the calculations of electron repulsion integrals(ERIs). For example, large-scale MO calculations with $N > 5,000$ spend longer than two months even on contemporary workstations. To perform such large-scale calculations in realistic time, some performance improvement technique, such as parallelization and hardware customization, should be considered.

In addition, to obtain total energy values within chemical accuracy of 0.01 kcal/mol, stricter numerical accuracy is required as the basis size increases. For example, in molecules with 10,000 basis, such as proteins, numerical accuracy for total energy values must be retained to at least 11 digits (i.e., the order of 1.0D-10) to keep accumulation of numerical errors less than the chemical accuracy (0.01 kcal/mol). To achieve this criterion, both some partial summation technique which would be utilized on parallel machines and longer mantissa (> 52 bits of double-precision real number) in floating-point number presentation are required[7].

For the reasons mentioned above, we cannot help recognizing that the *ab initio* MO method is not useful enough to analyze the process of biological and chemical phenomena and to design materials and medical molecules based on such analysis. For further development of material science, super high-performance and cost-down of *ab initio* MO calculations would be inevitably be needed.

We have been therefore developing a special-purpose parallel computer *MOE (Molecular Orbital calculation Engine)*, so as to leverage the cost/performance of *ab initio* MO calculations over the current high-performance computer systems[8]. MOE consists of multiple processing nodes, each of which is made of an MO-specific PPRAM (Parallel-Processing RAM) chip and some external memory. Each PPRAM chip provides (1) a general-purpose RISC-type integer processor, (2) an ERI-specific 76 bit floating-point MULTIPLY-and-ADD processor, (3) a standard inter-chip communication interface (PPRAM-Link interface), and (4) on-chip SRAM memory.

Figure 16 illustrates the overview of the MOE system.

B. Architecture of PPRAM-MOE

As previously stated, all PPRAM-MOE chips have a 76 bit data-width floating-point MULTIPLY-and-ADD unit, which is called FU. The detail of operations at MOE processing nodes are classified as follows.

- Floating-point MULTIPLY-and-ADD calculations : '$ERI0 = ERI0 + coefficient \times ERI1$' on the large-size 76 bit-width memory space, in order to execute a ERI-calculation program. The *ERI0, 1* are intermediary integral values.
- Normal floating-point calculations, in order to execute the calculations of initial value, Fock matrix elements, etc.
- Communication control via PPRAM-Link, integer calculations, etc.

The ERI-calculation program is a sequence of FU operation instruction set, and it is characteristic in that there is low reference dependency in the recursive formula of ERIs. Therefore, the direct memory-to-memory operand instruction set is suitable

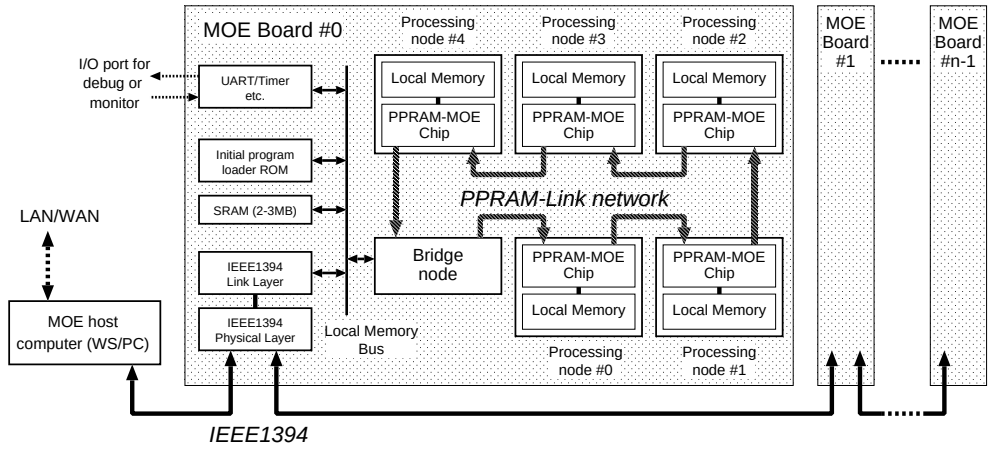

Fig. 16. Overview of MOE System

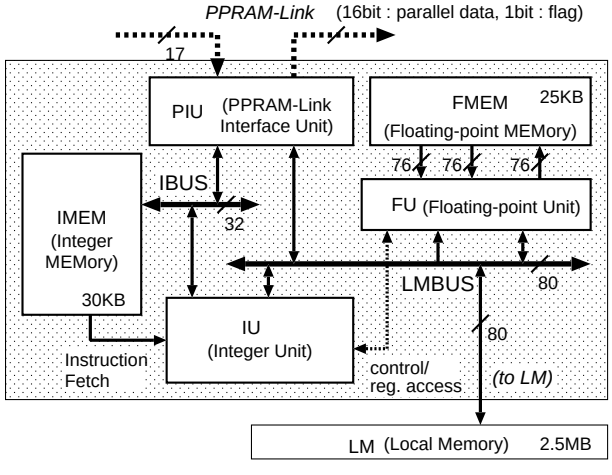

Fig. 17. Organization of MOE Processing Node

for FU. And in order to execute integer/control operations effectively, general integer processing unit (IU) is provided, apart from FU. The internal structure of a processing node is illustrated in Figure 17.

A processing node is constructed by a node core chip, which is called "PPRAM-MOE chip", and a local memory (LM) of 2.5MB size and 80bit-width. FU,IU and PPRAM-Link interface unit (PIU) are embedded in PPRAM-MOE chip. Moreover, IMEM (integer memory) of 30KB size, 32bit-width, and FMEM (floating memory) of 25KB, 76bit-width are embedded in it. The control program and integer data are stored to IMEM, floating point data are stored to FMEM.

All of data for calculations are transfered from the MOE host computer, and these are stored to LM. The access latency time is longer than IMEM and FMEM, because LM is general SRAM chip. However, there is no branch instructions in the ERI-calculation program, therefore we adopt to conceal LM's access latency time by reading two FU instructions per one LM read cycle. Hence in order to speed-up FU calculations, we let FU instruction-set to be 40 bit-width, and moreover we restrict to assign LM addresses to both destination and source operand.

The summary of specifications of IU,FU and PIU is shown as follows.

IU : 32 bit-width pipelined RISC processor with internal integer registers. IMEM is used for the instruction and data memory. IU also access LM and FMEM freely for integer calculations. IU can control FU directly, to operate floating-point operations which is not rely on the ERI-calculation program.

FU : 76 bit-width pipelined floating-point MULTIPLY-and-ADD operation unit with internal floating-point registers of 76 lines. This sequential calculation according to the ERI-calculation program is independent from IU, but its start/end pointer address and solitary floating-point operations are in control of IU.

PIU : Interface unit based on PPRAM-Link standard. PIU is used for communication between five processing nodes and a bridge node in MOE board. PIU consists of the logical/physical layer, the logical layer circuit includes a direct memory access controller(DMAC) and a transaction processing logic. Therefore, MOE host can read/write whole IMEM and LM at all of processing nodes. The framework of PIU logical layer is shown in Figure 18, besides functions of the logical layer, the skew adjustment on 17 bit-width parallel lines is provided. At current MOE system, the bandwidth allocation protocol and some transactions:"Move, Event, Active-message" are omitted, because they are unnecessary. Hence, an amount of PIU transaction logic can be reduced. And, CSR is also disregarded, because of the restriction of MOE architecture.

C. Performance estimation

We had an analysis about the relation between cumulative calculation time and required communication time for Gramicidin (11910 basis functions), assuming that all of required ERIs are computed by single MOE processing node. The analysis says that it will be 1.56×10^8 seconds (about 1800 days) to execute all ERIs, and required communication bandwidth peak will be 0.6 Mbps.

If assuming MOE system with 100 processing nodes, it will be required about 60 Mbps ($= 100 * 0.6$) for the bandwidth

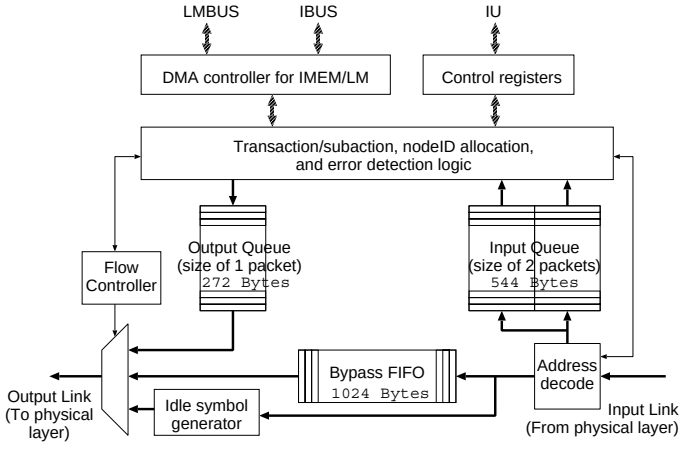

Fig. 18. Organization of PPRAM-Link I/F Unit (PIU)

TABLE II
EFFECTIVE COMMUNICATION PERFORMANCE IN MOE BOARD

Transaction type	Read 256B	Write 256B
Time for 1 transaction [μs]	9.66	9.62
Effective bandwidth [MB/sec.]	26.5	26.6

PPRAM-Link transmission cycle = 50 MHz, 17 bit-width.

bottleneck connection between MOE host and MOE boards. But it will be only 3.0 Mbps per one MOE board, there is no problem with regarding to PPRAM-Link communication performance. Table II shows effective communication performance and time required for one transaction between PPRAM nodes in same MOE board, on condition that the bridge node reads/writes IMEM of 5 PPRAM-MOE nodes continuously by transactions with 256 Byte payload-size.

In case of increasing the number of processing nodes of MOE system, all ERI calculations will be ideally distributed to each processing nodes. Besides, time required for inter-node communications is quite little then ERIs calculation time.

If MOE system with large number of processing nodes targets large molecules such as Gramicidin, the communication delay can be concealed in calculation processing time, and ERI calculation tasks will be ideally distributed to each processing nodes.

V. PLIF CORE

In order to do verify performance of PPRAM-Link and to promote it, general-purpose PPRAM-Link logical layer logic and its estimation environment, are needed. "PPRAM-Link I/F (PLIF) Core" is a group of logic IP (Intellectual Property) core based on PPRAM-Link logical layer standard (version 1.0) and additional interface logic. PLIF Core is written in generic VHDL, so it is useful for the logical simulation and implementations in FPGA/ASICs.

Basic block of PLIF Core is called PLIF Kernel. It is useful to estimate logical functions of PPRAM-Link itself, and fundamental communication simulations. Figure 19 illustrates the framework of PLIF Kernel. It has ringlet interface to input/output 17 bit-width symbol, and input/output queue interface. Control logic in PLIF Kernel have function/protocols as follows.

- Subaction-level communication control
- Node ID assignment at single-ringlet
- Bandwidth allocation
- Primitive error detection

PLIF Core is written and distributed under the GNU General Public License which means that its source code is freely-distributed and available to the general public. The following lineup is provided now, and several packages can be obtained from our web site.

PCI bus edition : Kernel + interface logic for PCI-bus. Using the edition, it become possible to insert whole of its circuits in a FPGA on PCI board. At the implementation, two or more PCs are inter-connected by unidirectional parallel lines. We had functional verification test of PPRAM-Link logical layer, on versatile two PCs.

VSIA edition : Virtual Socket Interface Alliance(VSIA) tries to specify open standards that facilitate the mix and match of Virtual Components from multiple sources[12]. The VSIA edition is as same as PLIF Kernel functionally, and additional interface logic is added. PLIF VSIA edition is useful for various type of inter-IP connection.

IBM CoreConnect edition : CoreConnect bus architecture, provided by IBM, is interface standard to design IP/chips. It enable to design IPs with plug-and-play and re-usability.[13]. CoreConnect is compliant with VSIA guidelines, and many commercial or non-commercial IPs with CoreConnect interface, have been released. Hence, PLIF CoreConnect edition is based on the VSIA edition. By combining desirable existent IPs and it, various PPRAM chips can be designed easily.

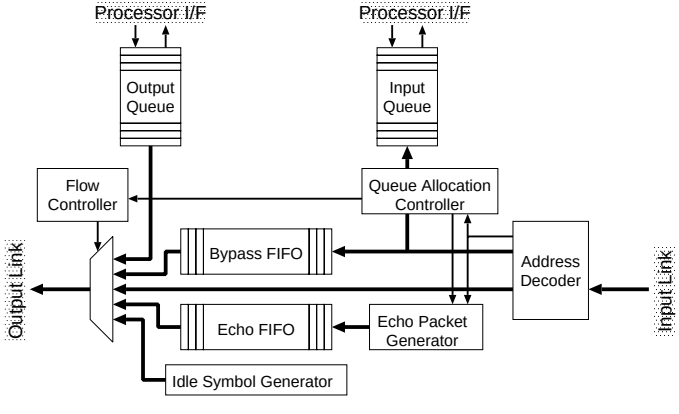

Fig. 19. Organization of PLIF Kernel

VI. CONCLUSIONS

In this paper, we proposed PPRAM definition and PPRAM-Link Standard Version 1.0 based on SCI. PPRAM is an architectural concept that build up computers or electronic equipment systems. And PPRAM-Link targets inter-chip connection in PPRAM-based system. PPRAM-Link standard consists of logical/physical layer and API interface, and we showed referential circuit design of the physical layer.

Following this, we outlined an example of PPRAM-based system, or MOE. MOE processing node has PPRAM-Link in-

terface unit (PIU), and PIU enabled to communicate automatically using PPRAM-Link transaction. Finally, we introduced PPRAM-Link interface IP, or PLIF Core. It is composed of Kernel Core, PCI bus, VSIA and CoreConnect editions.

In recent years, some serial interconnection standard with one or more Gbps bandwidth, are being appeared. They targets mainly inter-chip and inter-MCM connections, and they will be useful as alternative PPRAM-Link physical layers. But when we try to use above-mentioned standards, the circuit cost overhead, especially of logical layer, will be more serious problem. So, we are considering new PPRAM-Link logical layer standard that aims following characteristics : (1) Logical compatibility to current PPRAM-Link, (2) Circuit reduction by sharing several protocol logic and segmentation inside protocol stacks, and (3) Higher communication performance and lower cost than conventional crossbar switch connection.

REFERENCES

[1] Wulf, W.A. and McKee, S.A.,"Hitting the Memory Wall: Implications of the Obvious," ACM SIGARCH Computer Architecture News, vol.23, no.1, pp.20–24, Mar. 1995.

[2] Wilkes, M.V., "The Memory Wall and the CMOS End-Point," ACM SIGARCH Computer Architecture News, vol.23, no.4, pp.4–6, Sep. 1995.

[3] Johnson, E.E., "Graffiti on "The Memory Wall"," ACM SIGARCH Computer Architecture News, vol.23, no.4, pp.7–8, Sep. 1995.

[4] 'Murakami, K., "Invited Talk: Current Status of PPRAM," 6th International Conference on VLSI and CAD (ICVC'99), Seoul, Korea, Oct. 26-27 1999.

[5] Horowitz, M., Yang. C.K.K., Sidiropoulos, S., "High-Speed Electrical Signaling: Overview and Limitations," IEEE Micro, pp.12-24, Jan./Feb. 1998.

[6] Miyajima, H., Inoue, K., Kai, K., and Murakami, K., "On-Chip Memorypath Architectures for Parallel Processing RAM (PPRAM)," Workshop on Mixing Logic and DRAM, Denver, Colorado, June 1997.

[7] Takashima, H., Kitamura, K., Tanabe, K., and Nagashima, U., "Is large-Scale *ab initio* Hartree-Fock calculation chemically accurate? Toward improved calculation of biological molecule properties,", Journal of Computational Chemistry Vol.20, No.4, pp.443–454, 1999.

[8] Hashimoto, K., Tomita, H., Inoue, K., Metsugi, K., Murakami, K., Inabata, S., Yamada, S., Miyakawa, N., Takashima, H., Kitamura, K., Obara, S., Amisaki, T., Tanabe, K., Hayakawa, K., and Nagashima, U., "MOE: A Special-Purpose Parallel Computer for High-Speed, Large Scale Molecular Orbital Calculation," Supercomputing (SC99), Portland, OR, Nov. 13-19, 1999.
`http://www.sc99.org/proceedings/papers/hashimot.pdf`

[9] Murakami, K., Shirakawa, S., and Miyajima, H., "Parallel Processing RAM Chip with 256Mb DRAM and Quad Processors", 1997 IEEE International Solid-State Circuits Conference, pp.228–229, Feb. 1997.

[10] PPRAM Project of Kyushu University,
`http://ppram.csce.kyushu-u.ac.jp/`

[11] PPRAM Consortium,
`http://www.ppram.or.jp/`

[12] VSI Alliance,
`http://www.vsi.org/`

[13] IBM CoreConnect bus architecture ,
`http://www.chips.ibm.com/products/coreconnect/`