

VLSI Architecture Based on Packet Data Transfer Scheme and Its Application

Yuya Homma and Michitaka Kameyama
 Graduate School of Information Sciences
 Tohoku University
 Sendai, Japan
 {yuya, kame}@kameyama.ecei.tohoku.ac.jp

Yoshichika Fujioka and Nobuhiro Tomabechi
 Department of System and Information Engineering
 Hachinohe Institute of Technology
 Hachinohe, Japan
 {fujioka, tomabech}@hi-tech.ac.jp

Abstract—Packet data transfer scheme is introduced for intra-chip data transfer to solve an interconnection problem. Double transmission lines are provided as a platform of the micronetwork. A protocol suitable for intra-chip data transfer is proposed to make a router as simple as possible. An application to a parallel VLSI processor is also discussed. In comparison with a multi-bus architecture the parallelism can be greatly increased under the same chip size because of the compactness of the micronetwork.

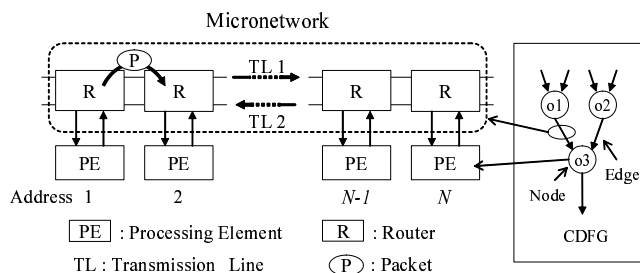


Fig. 1. Parallel processing architecture

I. INTRODUCTION

One of the most serious problems is performance degradation due to interconnection complexity in recent System-on-Chip (SoC) implementation [1]. On-chip physical interconnections will present a limiting factor for performance and, possibly, energy consumption.

The conventional multi-bus data transfer architecture requires a number of switches, which makes the chip area very large. This paper presents a packet data transfer scheme for intra-chip data transfer to solve the interconnection problem. A network on chip architecture has been already proposed for the purpose of IP interface between macro modules [2], [3], [4], [5]. The on-chip micronetwork will meet the distinctive challenges of providing functionally correct, reliable operation for interacting SoC components. However, protocol-based reduction of interconnection complexity and delay has not been reported until now.

The proposed micronetwork consists of double transmission lines and routers. Each router is directly connected to a processing element (PE). It is the most important factor to implement the routers as simple as possible for reduction of hardware complexity. After completion of a router-router transfer mode, we provide a PE-router transfer mode. Such a two-mode packet data transfer scheme makes free from the packet collision, so that we can design a very simple and high-speed router.

An application to a parallel VLSI processor is also discussed. The area of the proposed micronetwork is sufficiently small, while its performance is almost same as the conventional multi-bus data transfer architecture. Therefore, more hardware resources can be contained in a fixed chip size, and we can improve the total performance.

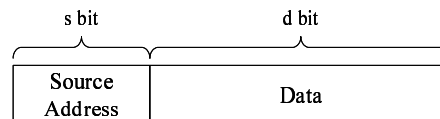


Fig. 2. Packet format

II. PACKET DATA TRANSFER ARCHITECTURE

In the following discussion, we consider a single-chip VLSI processor composed of multiple PEs connected by micronetwork as shown in Fig.1. Assume that a processing algorithm is given, and it is represented by a control data flow graph (CDFG), so that the scheduling and allocation can be determined in advance. That is, static scheduling and allocation are effectively employed for the parallel processing including data transfer. Each node in the CDFG is allocated to a PE, and inter-PE data transfer corresponding to an edge is done through the micronetwork.

On the micronetwork, there are two transmission lines used for the packet data transfer: one is used for the direction from left to right, the other is used for the direction from right to left. Bit-parallel packet data transfer is done on the micronetwork. The packet data transfer between two PEs can be done through the routers directly connected to the PEs.

As shown in Fig.2, a packet consists of a source address and a data field. Each router has a selection address which is used to determine whether the packet is received or not. That is, the packet is received by the routers having the selection address equal to the source address. Then, the packet is transferred to the PE directly connected to the router.

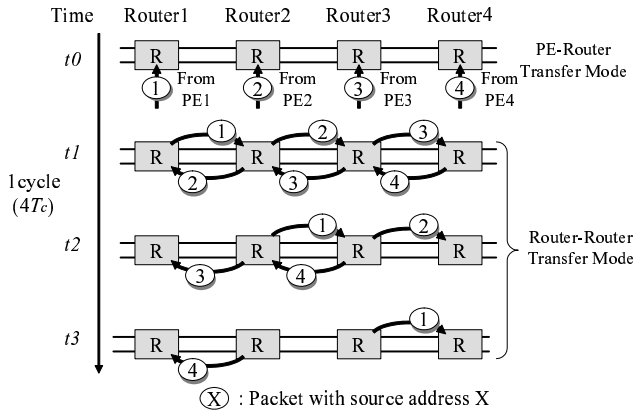


Fig. 3. Example of the packet data transfer

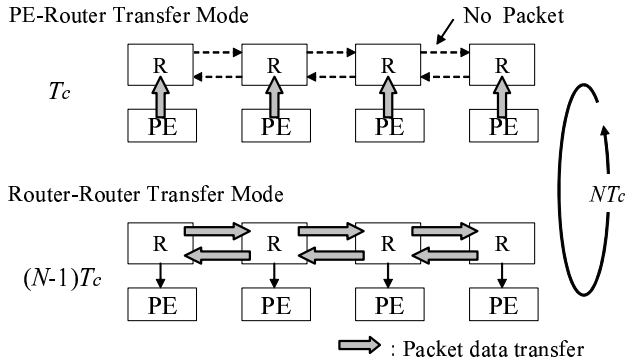


Fig. 4. Two kinds of transfer mode for packet collision avoidance

Figure 3 shows an example of the two-mode packet data transfer scheme in the micronetwork having four routers. In the PE-router transfer mode, each packet is transferred to a router at a clock cycle t_0 , where the clock cycle time is T_c . In the router-router transfer mode, each packet is transferred to an adjacent router in a pipeline manner without collision. At the worst case, it takes four clock cycles of $4T_c$. Then, every packet data transfer is completed, so that PE-router transfer mode can be started again. Generally, NT_c will be the packet data transfer time at the worst case in the micronetwork having N routers as shown in Fig.4.

Not only point-to-point packet data transfer but also parallel broadcasts can be done in the proposed micronetwork. Figure 5 shows an example of two parallel broadcasts in the micronetwork having four routers. A packet from PE1 is transferred to PE2 and PE3, and the other packet from PE3 is transferred to PE1 and PE4 in parallel.

Figure 6 shows the router structure for unidirectional packet data transfer. If the source address is equal to the selection address, Data is latched to Register1. The packet from Router $i-1$ is selected by multiplexers when the router-router mode is effective, otherwise the packet from PE i is selected. Two pipeline registers are also provided for pipeline transfer.

We can extend the above mentioned basic micronetwork into more complex ones. For an example, a hierarchically

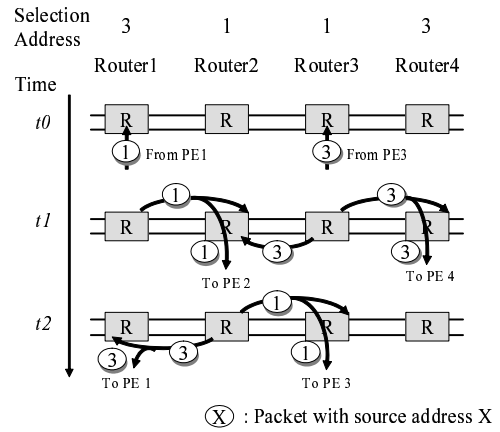


Fig. 5. Example of the parallel broadcast

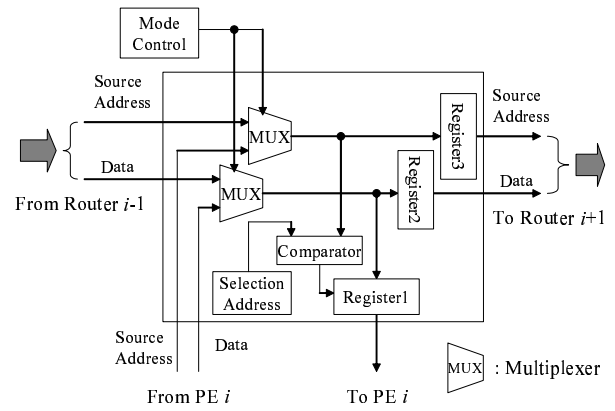


Fig. 6. Router structure

connected micronetwork is constructed using the proposed basic micronetwork of Fig.1 as a macro module as shown in Fig.7. Thus extended micronetwork gives tree structure, where PE is inserted between the macro modules at different levels. According to the specified scheduling determined in advance, the packets at a waiting state are stored temporally in the PE.

III. APPLICATION TO A PARALLEL VLSI PROCESSOR

As an application of the Packet Data Transfer Architecture (PDTA), we consider a VLSI processor for multi-operand multiply-additions[6]. The parallel VLSI processor structure is shown in Fig. 8. The PE consists of an adder, a multiplier and a local memory. We can select an operation mode from a multiplication, an addition and a memory access. The operations of the PEs and the routers are controlled by VLIW architecture.

The typical VLIW field is shown in Fig.9, where each router selection address and the number of clock cycles in a single step are specified. A step is a cycle when packet data transfer from a source to a destination and a PE operation are done.

Let us consider an example of the multi-operand multiply-addition shown in Fig.10. In Step1, data transfer between PE2 and PE1, and data transfer between PE4 and PE3 are

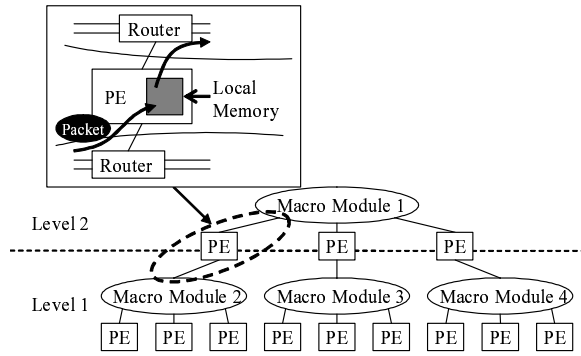


Fig. 7. Hierarchically connected micronetwork

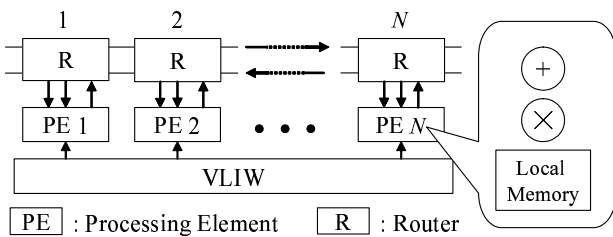


Fig. 8. Parallel VLSI processor structure

| | | | | |
|---------------------|---------------------|-----|---------------------|-----------------------------------------|
| Selection Address 1 | Selection Address 2 | ... | Selection Address N | Number of Clock Cycles in a Single Step |
|---------------------|---------------------|-----|---------------------|-----------------------------------------|

Fig. 9. Typical VLIW field

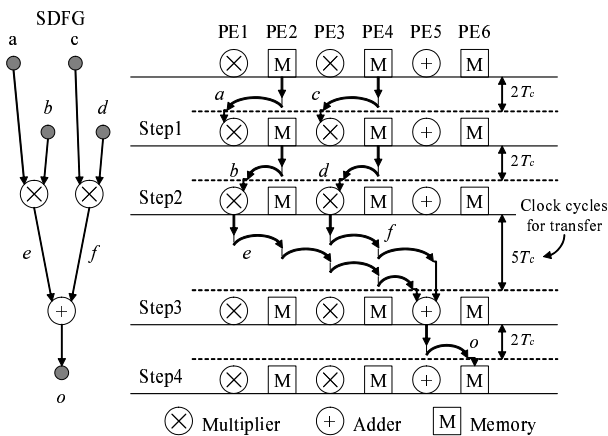


Fig. 10. Example of scheduling and allocation

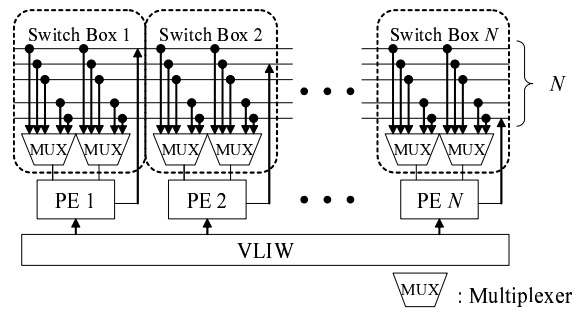


Fig. 11. Parallel VLSI processor using multiple buses

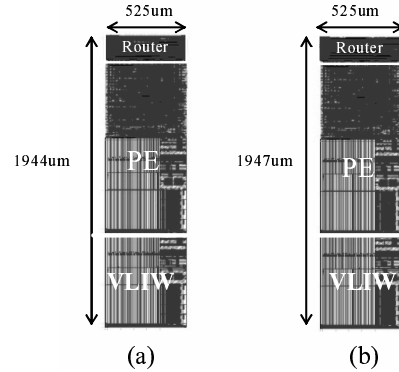


Fig. 12. Layouts of modules composed of a PE and a router (a)Router for 3-bit address packet data transfer between 8 PEs, (b) Router for 5-bit address packet data transfer between 32 PEs

simultaneously done for the time $2T_c$. Then, memory access of Data b and Data d are done. In Step3, data transfer between PE1 and PE5, and data transfer between PE3 and PE5 are simultaneously done for the time $5T_c$. Then, addition of Data e and Data f are done. Thus, packet data transfer times are different each other according to transfer distance.

Figure 11 shows a parallel VLSI processor based on the conventional Multi-Bus Data Transfer Architecture (MBDTA). A switch box composed of multiplexers is used to control the connection between PEs. The area of the interconnection network is increased because a number of bus switches are required to program the specified data transfer.

The 32-bit VLSI processor chips are evaluated using $0.18\mu\text{m}$ CMOS design rule. Figure 12 shows layouts of modules composed of a PE and a router in PDTA. Table I shows future of a PE. The number of packet address bits is given by $\log_2 N$, where N is the number of PEs which can be integrated on a single chip. Therefore, the increase of N does not make the area of the module for packet data transfer between N PEs so large.

Figure 13 shows layouts of modules composed of a PE and a switch box in MBDTA. The number of buses is given by N in the multi-bus architecture. Therefore, the increase of N makes the area of the module for data transfer between N PEs very large.

Table II shows the result of performance evaluation, where

TABLE I
FEATURE OF A PE

| | |
|----------------------|---------------------------|
| Design rule | 0.18 μ m CMOS |
| Word length | 32 bit |
| Multiplier | Wallace-tree multiplier |
| Adder | Carry lookahead adder |
| Local memory | 32 bit \times 256 words |
| Control memory field | 32 bit \times 256 words |

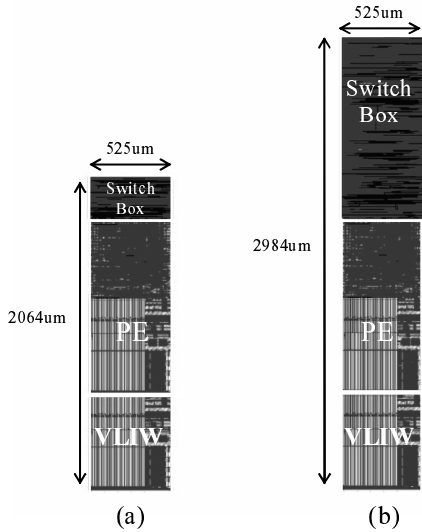


Fig. 13. Layouts of modules composed of a PE and a switch box (a) Switch box composed of 8-to-1 multiplexer for data transfer between 8 PEs, (b) Switch box composed of 32-to-1 multiplexer for data transfer between 32 PEs

TABLE II
PERFORMANCE EVALUATION OF PARALLEL PROCESSORS

| Chip Size (mm^2) | PE Number N per Chip | | Step Cycle Time T_s (nsec) | | Ratio of Throughputs (FP/FM) |
|----------------------|------------------------|-------|------------------------------|-------|----------------------------------|
| | MBDTA | PDTA | MBDTA | PDTA | |
| 50 | 32 | 49 | 24.2 | 34.8 | 1.07 |
| 441 | 128 | 430 | 53.0 | 109.4 | 1.63 |
| 5,616 | 512 | 5,459 | 168.2 | 449.8 | 3.99 |

TABLE III
PERFORMANCE EVALUATION OF NETWORKS

| PE | MBDTA | | | PDTA | | |
|-----|----------------|------------|----------------------------|----------------|------------|----------------------------|
| | A (mm^2) | T (nsec) | AT ($mm^2 \cdot nsec$) | A (mm^2) | T (nsec) | AT ($mm^2 \cdot nsec$) |
| 32 | 21 | 9.6 | 202 | 3.2 | 21.4 | 69 |
| 128 | 330 | 38.4 | 12,672 | 13.0 | 96.2 | 1,251 |
| 512 | 5,276 | 154 | 810,394 | 52.6 | 437 | 22,981 |

N is the maximum number of PEs which can be integrated in each chip size. Assume that times for a multiplication, an addition and a memory access are equal each other. During the step cycle time T_s , data transfer between PEs, and either of a multiplication, an addition and a memory access are done as shown in Fig.10. The throughput FP of PDTA is defined to be the number of steps executed per second. Hence, $FP = N/T_s$. We can similarly define the throughput FM of MBDTA.

Under the same chip area constraint, the ratio of the throughputs FP/FM is evaluated. It is clear that the performance of PDTA is superior to that of MBDTA. The packet data transfer time is increased in proportion to the number N of PEs, so is the multi-bus data transfer time. The area of micronetwork is proportional to N . While the area of multiple buses is proportional to N^2 , because both the area of a switch box and the number of switch boxes are proportional to N .

Let us evaluate only the interconnection resources for data transfer such as switch-boxes and micronetwork lines. Table III shows comparison between PDTA and MBDTA, where A is an area of the interconnection resource and where T is a data transfer time. If the same number of PEs are contained in the parallel processor, the AT product of PDTA is much smaller than that of MBDTA. This means that high-throughput data transfer can be achieved on smaller interconnection resources.

In future CMOS LSI, the interconnection delay will become much larger than the delay of active devices. Therefore, the interconnection delay becomes dominant rather than the router delay in the packet data transfer. This implies that the advantage of the packet data transfer scheme will be more evident because packet data transfer time can be almost same as the multi-bus data transfer time.

IV. CONCLUSION

Very simple packet data transfer architecture is developed to solve a SoC interconnection problem. Because arbitrary data transfer can be done by programming selection addresses, the flexibility of the data transfer scheme will give a solution for the interconnection problem also in dynamic reconfigurable VLSI processors. Thus, the new concept of packet data transfer will open up a new System-on-Chip technology.

REFERENCES

- [1] D. Sylvester and K. Keutzer, "A global wiring paradigm for deep submicron design," IEEE Trans. CAD/ICAS, Feb. pp. 242-252, 2000.
- [2] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," In Proceedings of the Design Automation and test in Europe, pp. 250-256, March 2000.
- [3] M. Mizuno, W. J. Dally and H. Onishi, "Elastic interconnects: repeater-inserted long wiring capable of compressing and decompressing data," in Proceedings of the IEEE International Solid-State Circuits Conference, pp. 346-347, 2001.
- [4] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in Proceedings of the 38th Design Automation Conference, pp. 684-689, 2001.
- [5] L. Benini and G. D. Micheli, "Networks on chips: a new SoC paradigm," IEEE Computer, pp. 70-78, Jan. 2002.
- [6] Y. Fujioka, M. Kameyama and N. Tomabechi, "Reconfigurable parallel VLSI processor for dynamic control of intelligent robots," Proc. IEEE Computers and Digital Techniques, vol.143, pp. 23-29, Jan. 1996.