Solutions for Real Chip Implementation Issues of NoC and Their Application to Memory-Centric NoC

Donghyun Kim, Kwanho Kim, Joo-Young Kim, Seung-Jin Lee and Hoi-Jun Yoo
Semiconductor System Laboratory, Department of Electronic Engineering and Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Republic of Korea
donghyun53@eeinfo.kaist.ac.kr

Abstract – **This paper describes real chip implementation issues of Network-on-Chip (NoC) and their solutions along with series of chip design examples. The solutions described in this paper cover both architectural aspects and circuit level techniques for practical chip implementation of NoC. As for architecture level solutions, topology selection, chip-aware protocol design, and On-Chip Serialization (OCS) for link area reduction are explained. For circuit level techniques, SERDES and synchronizer design, crossbar switch partial activation, and low-voltage link are presented as the foundations for power and area efficient NoC implementation.**

Regarding presented solutions for NoC implementation, this paper proposes memory centric NoC (MC-NoC) for homogeneous multi processor SoC (MPSoC). Flexibility and feasibility of task mapping on homogeneous SoC are the key feature of the MC-NoC. 8 dual port SRAMs connected to crossbar switches in hierarchical star topology network facilitate data communication between processors, regardless of task mapping into the MC-NoC. Experimental result obtained by mapping edge detection tasks on the MC- NoC in various configurations shows almost constant performance. This result proves the effectiveness of the proposed architecture. The MC-NoC based SoC is also implemented on TSMC 0.18 um process technology.

I. Introduction

Networks-on-Chip (NoC) is emerging as promising technique for future complex SoC which consists of more than few tens of Intellectual Properties (IPs). Modular structure of NoC makes chip architecture highly scalable, and well-controlled electric parameters of the modular block improve reliability and operation frequency of on-chip interconnection network. After the NoC design paradigm has been proposed [1, 2], research area concerning NoC is fairly matured. There have been many researches on theoretical aspects of NoC such as topology selection, quality of service (QoS) guarantee, automated design tool development and routing scheme design [3-6]. Although these researches improve knowledge and facilitate high-level design of NoC, noble circuit techniques essential for real chip implementation haven't been covered in detail. In addition to that, importance of the study focusing on the real chip implementation is emphasized along with multi-processor SoCs (MP-SoCs) are emerging on market, such as Intel Quad Processor [7].

In this paper we describe design issues come up with real chip implementation of NoC and their solutions in the perspective of architecture and circuit level techniques, based on the chip design examples of authors' research group [8-12]. Architecture level issues of NoC include selection of optimal NoC topology and designing communication protocol regarding hardware complexity. For practical chip implementation, these issues are discussed mainly concerning power consumption and silicon area required. Evaluation on various topologies of NoC shows that hierarchical star topology is the most efficient in general. Then, effect of serialization ratio on the interconnection link and switch fabric area is also discussed. Finally, NoC protocol which adopts aligned packet format for hardware simplicity is explained. As for the circuit level techniques of practical NoC implementation, high speed SERDES, synchronizer design, low-voltage swing link, and crossbar switch partial activation technique are described in later section.

Based on the experiments and analysis presented in this paper, we propose memory centric NoC (MC-NoC) for homogeneous multi processor SoC. The MC-NoC is comprised of 8 dual port memories and crossbar switches configured in hierarchical star topology, providing dynamic communication channels between processors. In contrast to conventional 2D mesh array NoC such as RAW microprocessor [14], the MC-NoC supports flexible mapping of task regardless of data communication characteristic between processors. Experimental results show that different task mapping configurations on the MC-NoC do not present much difference in overall performance. This result proves flexibility and feasibility of task mapping on the proposed MC-NoC.

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2006-(C1090-0603-0012)

The rest of the paper is organized as follows. In section 2, previous works in authors' research group and other related works are briefly explained. After that, chip implementation issues of NoC and their solutions in the perspective of architecture and circuit level technique are presented in section 3. Section 4 proposes the MC-NoC regarding the solutions discussed in section 3. After the architecture and operation of the proposed MC-NoC is explained in detail, benefits of applying MC-NoC to the homogeneous multi-processor SoC are also described. Experimental results obtained by mapping edge detection task on MC-NoC are given in section 5. Finally, summary of implementation results and conclusion are made in section 6 and 7 respectively.

II. Related Works

As mentioned in the previous section, many researches on theoretical aspects of NoC have been reported. For high-level design feasibility of NoC, SUNMAP & Xpipes tools are recognized as automated tools for topology selection and synthesis, respectively [3][5]. In the SUNMAP tool, best topology of NoC is automatically selected for a given bandwidth constraint. Then, Xpipes generates synthesizable RTL for the selected NoC topology. Other researches on high-level design of NoC include AEthereal and DyAD NoC. In the AEthereal NoC, traffics are divided into two classes – guaranteed service (GS) and best effort (BE) – and combined router for both classes of traffic was designed to resolve quality-of-service (QoS) problem in NoC [4]. The DyAD NoC was proposed combining the advantages of deterministic and adaptive routing to improve performance and capacity of NoC [6]. Although, only few of the previous works are introduced here, vast amount of high-level design and theoretical studies of NoC are being actively performed by numerous research groups.

On the other hand, researches performed by authors' group have been focused on circuit level design issues of NoC regarding real chip implementation. The world first chip implementation of packet switched NoC was featured with on-chip serialization (OCS) and strobe signal based synchronization [8, 12]. The OCS technique which is enabled by the proposed synchronization scheme reduces silicon area required for interconnection link and switch fabric. Circuit techniques for low power NoC were also proposed along with heterogeneous SoC implementation [9, 11]. In this work, low-voltage signaling, crossbar switch partial activation, and SILENT coding schemes for serial link are presented as the circuit techniques for low power NoC. The most recent chip was designed aiming at high speed NoC. In this work, wave front train (WAFT) SERDES and aligned packet format are proposed [10]. The aligned packet format reduces hardware complexity of network interface modules while WAFT SERDES implements high speed serial link for high performance NoC. In addition to that, crossbar switch with adaptive bandwidth control was also proposed to lower the operation frequency of NoC without loss in bandwidth [13]. Beside the chip implementations of NoC, analytical studies also have been performed in authors' research group. In the analytical work for low power NoC design, various NoC topologies were compared in the point of power consumption and silicon area required [11]. And, impact of serialization ratio and protocol design on power consumption and chip area was also discussed to answer the practical chip implementation issues [12]. In the next section, we will discuss chip implementation issues of NoC and their solutions based on the experiences obtained from the previous researches.

III. NoC Implementation Issues & Solutions

This section discusses chip implementation issues of NoC and corresponding solutions in architectural aspect and circuit level techniques. Architecture level issues of NoC discussed in this section are topology selection, protocol design and serialization scheme design. The circuit level issues include high speed SERDES, synchronizer design, crossbar partial activation technique and low-voltage swing link.

A. *Topology Selection*

One of the challenging issues in the NoC design is choosing the best topology to meet the bandwidth and latency requirements for the target application with the lowest power and area cost. In these prior works, the candidate pool of topologies was limited to the regular and homogeneous topologies like a mesh, torus, cube, tree or multistage network. However, in the heterogeneous SoCs like embedded systems, the communication flows are certainly localized, not uniformly distributed. In this case, it is highly possible that the optimal topology can be a heterogeneous and hierarchical topology rather than a homogeneous and flat topology. In this section, such hierarchical and heterogeneous network topologies are investigated briefly in practical terms of energy consumption and area costs.

Topologies are categorized into two groups in this analysis: flat topologies such as a bus, star, mesh, and point-to-point and hierarchical topologies, for example *local-bus global-star, local-star global-mesh* and *local-star global-star*. The hierarchical topologies consist of a local and global network topology where the local and global network can have any type among the basic topologies. We comparatively analyze the four flat topologies and the three hierarchical topologies.

We assume that the size of each processing element (PE) is uniform as 1mm x 1mm and the PEs are placed as a square matrix regardless of the topology. The number of PEs, N, scales from 16 to 100. The hierarchical topology is assumed to be divided into $\sqrt{N}$ of clusters and each cluster contains $\sqrt{N}$ of PEs. There are two kinds of traffic patterns; one is uniform random traffic and the other is localized traffic with a locality factor. The locality factor means a ratio of the intra-cluster traffic to the overall traffic. In a hetero-

$$E_{pkt} = H_{Avg} \cdot (E_{Queue} + E_{ARB} \cdot SS_{Avg} + E_{SF} \cdot SS_{Avg}) + L_{Avg} \cdot E_{Link} + E_{Queue}$$

$$A_{Tot} = H_{Tot} \cdot (A_{Queue} \cdot SS_{Avg} + A_{ARB} \cdot SS_{Avg} + A_{SF} \cdot SS^2_{Avg}) + L_{Tot} \cdot A_{Link}$$

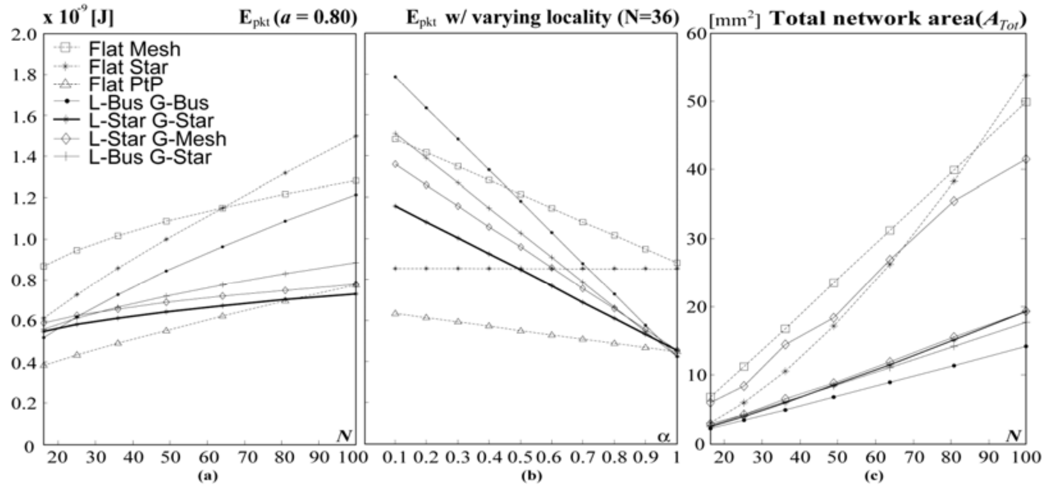


Figure 1. (a) Energy consumption according to the number of PEs and (b) traffic locality factor and (c) network area according to the number of PEs

geneous system, the locality factor can represent the localized traffic pattern quantitatively.

We use an average packet traversal energy E_{pkt} as a network energy efficiency metric which can be estimated by the equation shown in Fig. 1, summing up the energies on switching hops, links and a final destination buffer [1, 15]. The area cost of a network can be derived by the equation in Fig. 1, summing up the area of switches and links. Those energy terms and physical area of a queuing buffer, an arbiter, and a switch fabric are measured from the circuit implementation in 0.18 μm technology [9].

Fig. 1(a), (b) shows the comparison of the energy consumption under various traffic conditions. In any traffic condition, the point-to-point topologies show the best energy efficiency. If the point-to-point topologies cannot be adopted due to its infeasibility, the performance of star topologies is the best among the others. If *N* is fixed to 36 (See Fig. 1(b)), for instance, the flat star is the best for less localized traffic while the hierarchical star (L-star G-star) is the best for more localized traffic. The mesh always consumes 30~80% more energy than the hierarchical star does. Fig. 1(c) shows the area comparison of all of the topologies. The hierarchical bus topology shows the lowest area cost. However, local-bus global-star/mesh and local-star global-star/mesh topologies also occupies as little area as the hierarchical bus does. This is because the area of total network strongly depends on local networks rather than a global network. If we compare the area of a network with that of PEs, the H-star network consumes 20% of the PE-area but mesh consumes 50%.

According to our analysis, as the traffic gets localized, the energy cost of the mesh does not scale down as much as other hierarchical topologies do. Moreover, the area cost of the mesh is usually three times larger than that of other hierarchical topologies. As a result, the hierarchical star (local-star global-star) topology is the most cost-efficient and scalable topology for the heterogeneous systems where the traffic is localized. The energy cost is the lowest among hierarchical topologies and the area cost is also comparable with the hierarchical bus.

B. Packet Format and Protocol

In this sub-section, we discuss packet format and protocol issues for a real implementation of NoC. Because a NoC's data link, network layers should be implemented with hardware, a definition of packet format has to consider the physical channel structure. Typically, independently defined packet, flit and phit (physical digit) formats shown in Fig. 2(a) are used to support the concept of a layered architecture. However, such variable length packet formats cause a packet processing burden for a NoC. Misalignment among packet, flit, and phit formats makes packet parsing operations difficult.

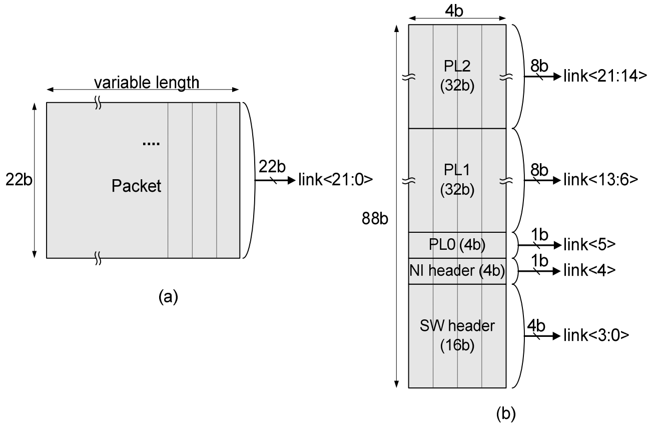


Figure 2. Packet format definitions: (a) typical and (b) aligned

For an efficient hardware implementation, we propose an aligned packet format which is defined in relation to the physical layer structure, shown in Fig. 2(b). This packet format defines a fixed-length packet and dedicated link wires are assigned to each packet field. Advantages of this scheme over the typical non-aligned packet format are as follows: The packet parsing process is very simple, and bit-width of a field can be easily increased with additional link wires. A disadvantage of this scheme is that its link utilization is inefficient if some fields are disabled. Although the non-aligned packet format can utilize the channel resource efficiently, the packet format has a complex packet parsing procedure and inflexible bit-width adjustment. As a result, the aligned packet format provides helps for efficient hardware implementation for NoCs.

C. On-chip Serialization

On-chip serialization (OCS) technique reduces the link width. Therefore the area and energy-consumption of the switch as well as the energy consumption of the link are also reduced. Fig. 3 illustrates the concept of the OCS. A serialization and de-serialization (SERDES) circuit are inserted at I/O of a processing unit (PU) reducing the bit-width of the I/O thus the switch size. The reduced crossbar switch size results in decrease of coupling capacitance of wires in the switch fabric, also contributes to the reduction of switch energy consumption. In the case of link, as the number of wires is reduced, wire space can be widen, which results in reduction of wire capacitance load or energy consumption. Because of the decreased bit-width, however, driver size must be increased to come up with increased operation frequency. In addition to that, the OCS increases switching activity factor because it breaks the correlation between consecutive transfer units which are often observed in the address field of memory access.

Fig. 4 shows the two conventional serializer circuits. The shift-register type serializer loads the parallel data through the 2:1 MUXs. After the load operation, the shift mechanism of the series flip-flops (F/Fs) realizes high-speed serialization. The maximum clock frequency is given by

$$f_{MAX} = \frac{1}{T_{MUX} + T_{SETUP} + T_{HOLD}},$$

where the T_{MUX} is the 2:1 MUX delay time, and the T_{SETUP} and T_{HOLD} are the setup and hold time of the F/F. The conventional architecture has two problems. First, maximum clock frequency is limited by the delay time of the D-FF. Second, the high-speed clock for the serialization becomes system overhead.

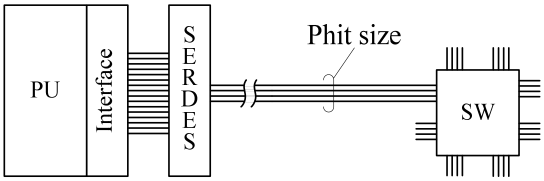


Figure 3. The concept of the on-chip serialization.

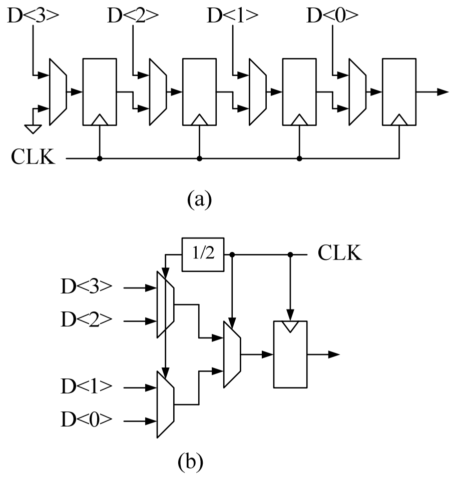


Figure 4. (a) Shift-register type and (b) MUX-tree type serializers

A new SERDES architecture, wave-front train (WAFT), is proposed to overcome these limitations [10]. The WAFT uses physical delay constant of delay elements (DEs) as a timing reference instead of clock, and utilizes signal propagation phenomenon instead of the shifting mechanism.

Fig. 5 shows a 4:1 WATF serializer circuit. When EN is low, D<3:0> is waiting at QS<3:0>. The VDD input of MUXP, which is called a pilot signal, is also loaded to QP. The GND input of MUXO discharges the serial output (SOUT) while the serializer is disabled. If EN is asserted, QS<3:0> and the pilot signal start to propagate through the serial link wire. Each signal forms a wave-front of the SOUT signal, and the timing distance between the wave-fronts is the DE and MUX delay which we call a unit delay. The series of wave-fronts propagates to the de-serializer like a train. When the SOUT signal arrives at the de-serializer, it propagates through the de-serializer until the pilot signal arrives at the end of the de-serializer, or STOP node. As long as the unit delay times of the sender and the receiver are the same, D<3:0> arrives at its exact position when the pilot signal arrives at the STOP node. When the STOP signal is asserted, the MUXs feed back its output to its input, so that the output value is latched.

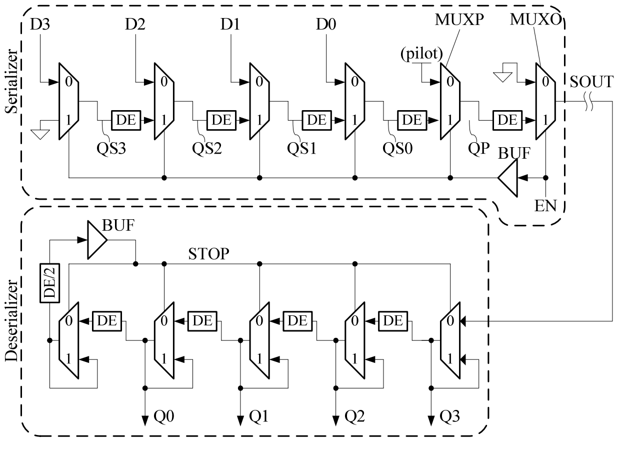


Figure 5. Wave-Front-Train Serdes

D. *Synchronization*

In multiple-clock-domain systems with heterogeneous PEs, the global synchronization among clock domains is getting challenging. One NoC contribution to such a SoC design is to remove the burden of global synchronization by using mesochronous communication, which implies that the network blocks share the same clock but the clock phase may different from each other due to asymmetric clock tree design.

One of simple and feasible ways to implement mesochronous communication is source synchronous technique in which strobe signal is transmitted together with data through a sideband link. Without global synchronization, the strobe signal is transmitted along with phits as a timing reference at a receiver. In order to suppress unnecessary power consumption, the strobe is activated only when a phit is valid. A receiving part latches the phit using the strobe signal, and synchronizes the phit with the local clock using a first-in-first-out (FIFO) synchronizer.

A design challenge in the source synchronous scheme is that the delay time for distributing the strobe signal over latches of the FIFO is significant, so that the strobe skew can possibly result in failure of latching the phit data. In order to solve this problem, matched-delay FIFO architecture is proposed [16]. Fig. 6 shows the matched-delay scheme in a receiver FIFO. The receiver FIFO has 12 20b flip-flops (F/Fs) and each 20b F/F is enabled by a clock signal generated by the ring counter. In the matched-delay architecture, the strobe is generated by the CLK_{NET} directly so that the strobe signal leads the phit signals by the amount of the BUFB delay time. As a result, both the strobe and phit have the same total delay time as

$$\text{Strobe delay} \approx \text{Phit delay} \approx t_{PD} + t_{BUFA} + t_{CQ} + t_{BUFB}$$

where the t_{PD} and the t_{CQ} mean propagation time on the long wire and clock-to-Q delay time of a F/F, respectively.

This architecture can be effectively applied to source synchronous scheme because it provides the matched delay regardless of the number of F/Fs, and operation frequency.

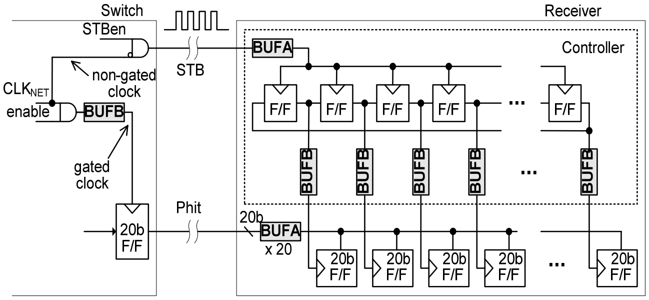


Figure 6. Matched-delay architecture in a receiver

E. *Low-power techniques*

1) *Low-voltage Signaling*

Global interconnect lengths can extend up to several millimeters in the case of a large SoC. Global links consume considerably more power compared to local links due to their large parasitic capacitance. Low-swing signaling can improve the power efficiency of such long wires significantly [11]. A low-swing interconnection is composed of a low-swing driver at the transmitting end and a sense amp at the receiving end. The driver uses V_{SWING} instead of VDD to drive the differential pair. Generally, lowering V_{SWING} decreases the power required to drive a long wire. However, V_{SWING} cannot be lowered indefinitely due to super-linearly increasing power consumption at the receiver side to restore the low-swing signal to its full-swing value. Instead, an optimum voltage swing level exists at which the energy consumed at the transmitting and receiving sides is minimized [17]. Careful simulation incorporating parasitic values extracted from layout can be conducted to determine this optimum voltage swing.

Fig. 7 shows a low-swing signaling scheme that has been implemented and tested on actual silicon [9]. The driving circuit is made using n-MOSFETs for both pull-up and pull-down transistors to take advantage of the lower linear resistance n-MOSFETs provide at small drain-source voltage compared to p-MOSFETs. A simple clocked sense amp is used with a three-stage inverter chain in the receiving circuit. The sense amp is designed to amplify differential input signals having as low as 200-mV swing to 1.6-V CMOS logic levels with small delay. A clock restoration circuit is used in conjunction with differential strobe signals to provide clocking for the sense amp. The circuit was designed with the transmitter and receiver connected by a 5.2mm wire. The wire was intentionally made in a winding pattern to simulate global interconnects in real SoCs. The delay was measured to be 0.9ns with a variation of less than 5%. The optimal voltage swing at a signaling rate of 1.6 Gb/s was found to be 0.3V. Thanks to the low-swing signaling, the power dissipation on the global link was reduced to just 1/3 of that on a conventional repeated link without the area overhead of using repeaters.

2) *Crossbar Partial Activation Technique*

A conventional n x n crossbar fabric is composed of n^2 crossing junctions that are implemented by NMOS pass-transistors. During each transaction, an input driver must charge or discharge 2n transistor-junction capacitors connected to the row-bar (RB) and column-bar (CB) that have been selected. As a result, the power consumption of the crossbar fabric becomes significant with a large number of ports. Unneeded power dissipation can be reduced using crossbar partial activation technique (CPAT) as illustrated in Fig. 8 [9]. By partitioning an n x n fabric into 4x4 tiles, the total activated capacitive loading is reduced by a factor of n/4. A gated input driver at each tile activates its sub-RB only when a column in that tile is selected by the scheduler. Only 4 additional four-input OR-gates are needed at each tile

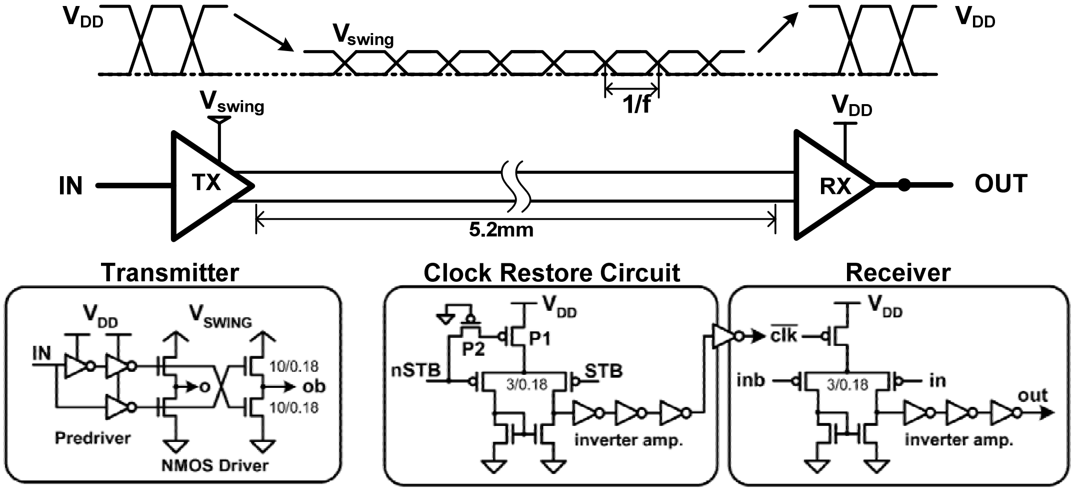


Figure 7. Low-signaling scheme and its transceiver circuits

for the implementation of the CPAT. The output path, CB, is also divided into two sub-CBs to reduce capacitive loading. The sub-CBs are joined at the output using a 2:1 MUX. For the 8x8 fabric shown, power savings of 22% is obtained at 90% offered load using the CPAT. The OR-gates and MUXs added account for only 2% of the overall power consumption. Applying the CPAT to a 16x16 fabric results in a 43% power saving.

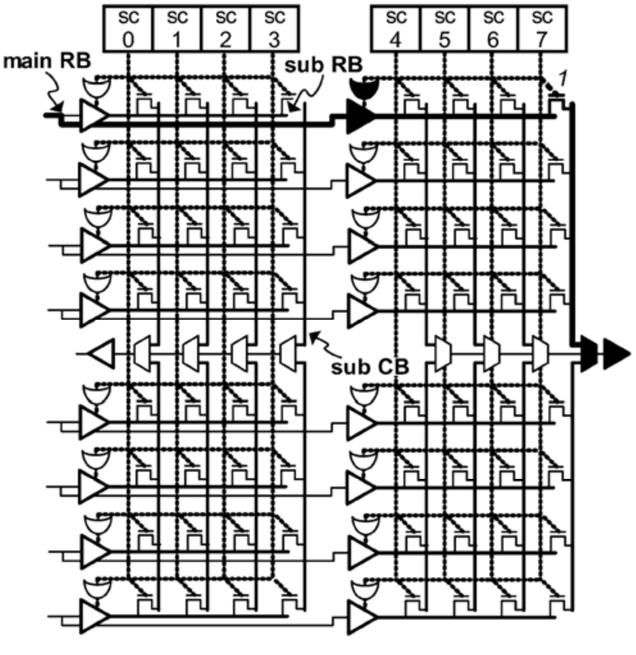


Figure 8. Proposed crossbar with partial activation

IV. Memory Centric NoC

In this section, we propose memory centric NoC (MC-NoC) which facilitates flexible and traffic-independent mapping of task on homogeneous MP-SoC. From the discussions in section 3, hierarchical star topology network, and synchronization scheme is adopted for the MC-NoC. Although, hierarchical star topology is proven to be advantageous for localized traffics of heterogeneous SoC, effectiveness of the hierarchical star topology still holds for MC-NoC which is applied for homogeneous SoC. This is because most of the traffics in homogeneous SoC are localized by applying the MC-NoC. The detailed description will be given later in this section.

1) Architecture & Operation

Fig. 9 (a) shows architecture and operation of the MC-NoC. In this figure, MC-NoC is applied for homogeneous multi-processor SoC which incorporates 10 RISC processors. Building blocks of the MC-NoC are dual port SRAM, crossbar switch, network interface (NI), and channel controller. In the MC-NoC, dual port SRAMs are dynamically assigned to the subset of the RISC processors involved in data communication. Then, shared data is exchanged by accessing assigned dual port SRAM. Crossbar switches of the MC-NoC provide non-blocking concurrent interconnections between dual port SRAMs and RISC processors. Operation frequency of the crossbar switches is decided to be twice of the other part of the MC-NoC to reduce overhead of packet switching latency. The NI performs packet processing and clock synchronization between crossbar switch and other building blocks of the MC-NoC. The key building block of the MC-NoC is the channel controller. The channel controller automatically manages communication channels between RISC processors to facilitate mapping of task on homogeneous SoC. Role of channel controller is described in more detail with the operation of the MC-NoC.

Fig. 9 (b) briefly represents important steps of MC-NoC operation. In this figure, crossbar switches are not drawn for simplicity of the description. While the operation is explained, we will assume that RISC processor 0 wants to pass the processed results into RISC processor 2 and 3. The MC-NoC operation is initiated by RISC processor 0 sending *Open Channel* request to the channel controller. The information about source and destination RISC processors is

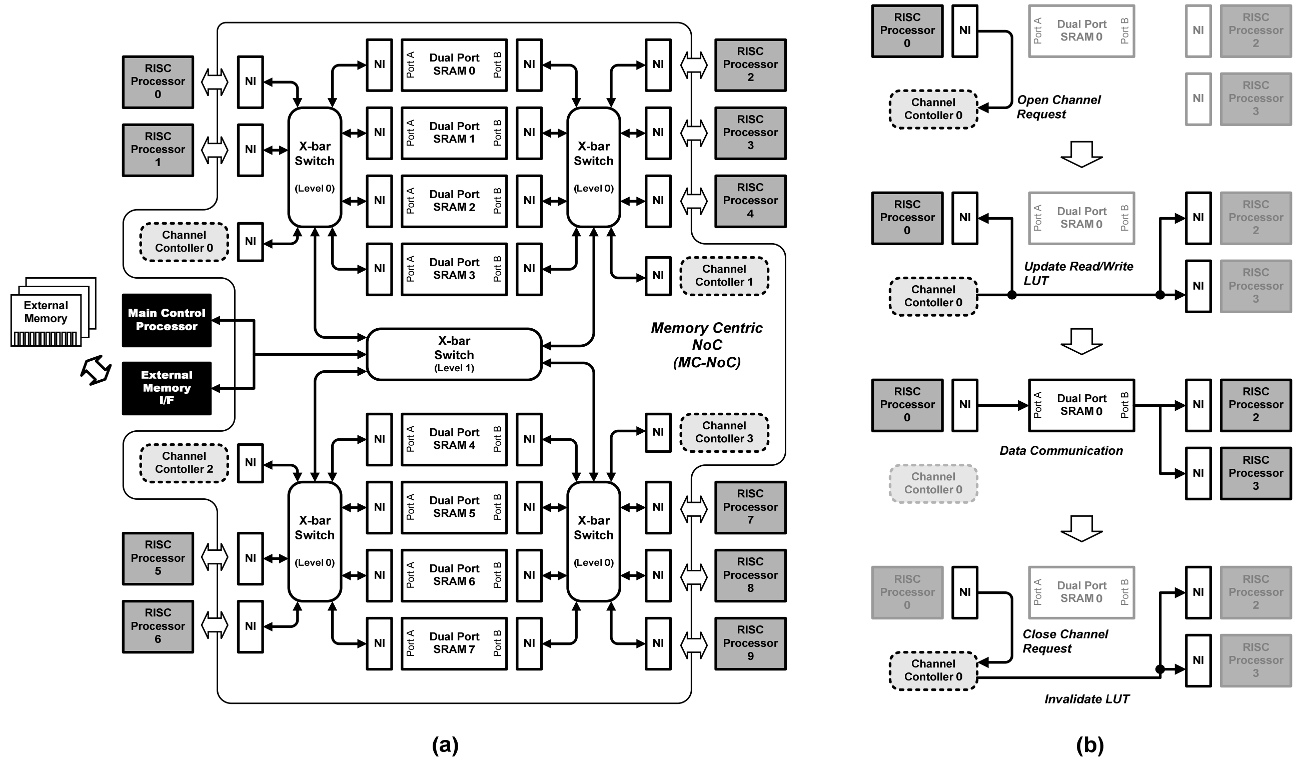


Figure 9. (a) Architecture of the MC-NoC, (b) Overview of the MC-NoC operation

also included in the *Open Channel* request. After that, channel controller assigns one dual port SRAM as a data communication channel if any of the SRAMs is available. By updating routing look up tables (LUT) in NIs of corresponding processors, SRAM assignment is completed. In this way, assigned SRAM is made to be accessible only for the RISC processors involved in data communication. At the end of data transfer through the dual port SRAM, source RISC processors send *Close Channel* request to the channel controller. Then, channel controller invalidates updated LUTs after checking completion of data transfer. In the proposed MC-NoC, each processor is able to send multiple *Open Channel* request as required. If all the SRAMs are used by other processors the data transfer should be stalled until one of the SRAM becomes available. In the MC-NoC *Open/Close Channel* request and LUT update are performed by sending special packets which is not visible to any processors or memories. By controlling operations of the MC-NoC using special packets, it has advantage of removing addition control signal wires.

While data communication is performed through the dual – port SRAM which is assigned by the channel controller, progress of data access from destination processors may differ from each other. To improve programming feasibility of the multiple RISC processors, the MC-NoC provides data synchronization scheme to resolve consistency problem occurred from the different data access order of destination processors. Fig. 10 (a) shows the case. Processor 2 reads data from address 0x0, while processor 3 accesses address 0xC. Until this moment, processor 0 has written valid data only at the address 0x0. The next step is shown in Fig. 10 (b). In this case, only processor 2 gets valid data from the dual port SRAM and processor 3 receives invalidate signal from valid check logic inside the dual port SRAM. After that, the NI of processor 3 holds the processor and retry read after specified wait cycles as shown in Fig. 10 (c). Once processor 0 writes valid data at address 0xC, processor 3 also gets valid data and continues processing. (Fig. 10 (d)) In the MC-NoC operation, the retry procedure described in Fig. 10 (c) is transparent to the RISC processors because the NI module of the MC-NoC automatically manages the procedure. As like *Open/Close Channel* request, invalid signal from valid check logic of the dual port SRAM is also transferred as a special packet. In our implementation, the valid check logic takes 5% of the dual port SRAM area. This is shown in Fig. 11.

2) Benefits of the MC-NoC

Main advantage of the MC-NoC is its flexibility of task mapping on homogeneous SoC. In this section the benefits of the MC-NoC is discussed through the comparison with conventional 2D mesh topology NoC.

As a task mapping example, edge detection operation is shown in Fig. 12 (a) [18]. In the figure, rectangular boxes represent processors performing tasks and solid/dotted lines

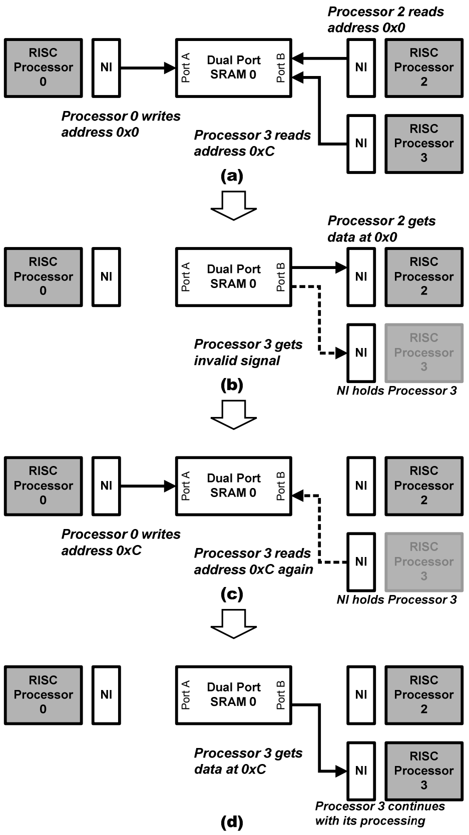

Processor 2 reads address 0x0
RISC Processor 0
NI
Dual Port SRAM 0
Port A
Port B
NI
RISC Processor 2
Processor 0 writes address 0x0
Processor 3 reads address 0xC
NI
RISC Processor 3
(a)
Processor 2 gets data at 0x0
Processor 3 gets invalid signal
NI holds Processor 3
(b)
Processor 0 writes address 0xC
Processor 3 reads address 0xC again
NI holds Processor 3
(c)
Processor 3 gets data at 0xC
Processor 3 continues with its processing
(d)

Figure 10. Data synchronization scheme of the MC-NoC

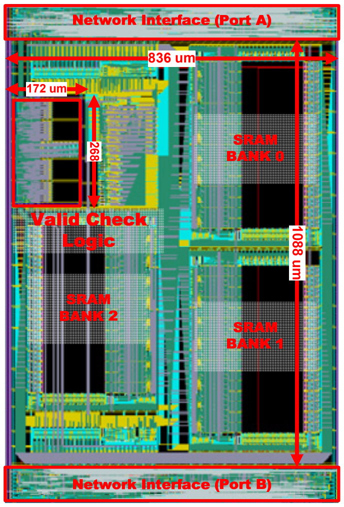
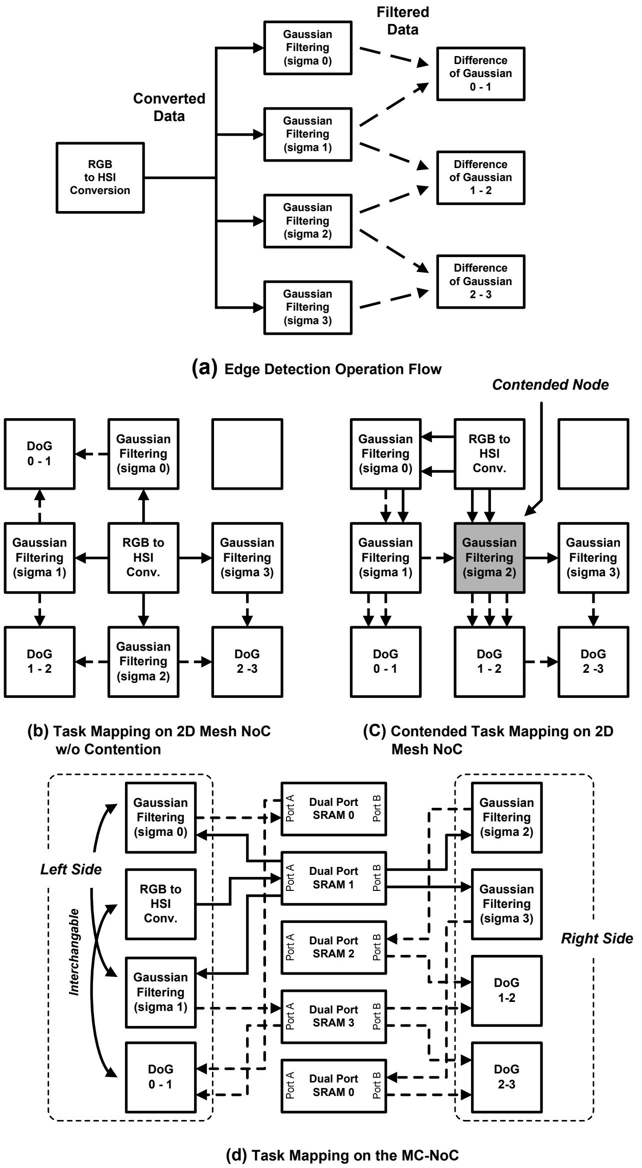


Figure 11. Layout of the dual port SRAM in the MC-NoC

depict data flow between processors. In this operation, input image is converted from RGB color space to HSI color space first. The converted image is processed by Gaussian filters with varying coefficients (sigma) and subtractions between filtered results are calculated to detect edges in different scale. Fig. 12 (b) and (c) shows mapping of edge detection operation on homogeneous SoC with conventional 2D mesh NoC. Because, there is no contention in data flow for the task mapping of Fig. 12 (b), it will outperform task mapping of Fig. 12 (c), even though all other conditions are equally given. The contention of data flow in Fig. 12 (c) is visualized by the number of arrows in the same locations. The drawback of the conventional 2D mesh NoC is dependency of overall SoC performance on mapping of the task. Even more, finding optimal task mapping may be very difficult for applications with complex data dependencies. Longer average hop counts and possibilities of deadlock on the way of finding badwidth-optimzed task mapping are additional drawbacks of the 2D mesh NoC.



(a) Edge Detection Operation Flow

(b) Task Mapping on 2D Mesh NoC w/o Contention

(C) Contended Task Mapping on 2D Mesh NoC

(d) Task Mapping on the MC-NoC

Figure 12. Mapping of Edge Detction task on conventional 2D mesh NoC and the MC-NoC

Feasibility of task mapping on the MC-NoC can be shown in Fig. 12 (d). For simplicity, crossbar switches are not drawn and only portion of the MC-NoC is depicted in this figure. In the MC-NoC, processors and dual port SRAMs are interconnected through the crossbar switches which provide full non-blocking connections. Therefore, interchanging task mappings just inside the left side or right side of the MC-NoC does not affect the data flow characteristic and resulting overall performance. For example, interchanging task mapping of difference of Gaussian (DoG) 0-1 and RGB to HSI conversion in Fig. 12 (d) has no impact on contentions in data flow. This attribute of the MC-NoC improves flexibility of task mapping on homogeneous SoC, because the key decision of the task mapping reduces to whether given task is mapped on the left side or right side. Dual port SRAM is adopted to remove performance loss when SRAM accesses are come from both left and right side of the MC-NoC.

In addition to that, variations on required bandwidth between co-working tasks are also successfully supported by the MC-NoC. If large bandwidth is required for some of tasks, multiple numbers of SRAMs can be dynamically assigned for the demanding task. For small amount of data transfer, only one dual port SRAM is assigned. In the point of traffic characteristic, the MC-NoC improves locality because most of packet transaction between processors and memories are confined into single crossbar switch to which the involved processors and SRAMs are connected.

V. Experimental Result

To demonstrate feasibility and flexibility of task mapping on the MC-NoC, this section briefly reports experimental results, showing how the overall performance is affected by different task mappings on the MC-NoC. For comparison, tasks of edge detection operation are mapped into to the homogeneous SoC shown in Fig. 9 (a).

The different mapping configurations are depicted in Fig. 13. Although the MC-NoC is drawn as single rectangular black box for simplicity, the architecture shown in Fig. 9 (a) is exactly applied for the performance comparison. At first, RGB to HSI conversion, Gaussian filter operations, and differences of Gaussian (DoG) calculation tasks are mapped randomly on the given architecture (Fig. 13 (a)). In the second, Gaussian filter operations are mapped on the upper half of the SoC, while DoG calculations are mapped on the lower half (Fig. 13 (b)). Similarly, the DoG and Gaussian filtering tasks are separated into left and right respectively, in the third task mapping configuration (Fig .13 (b)). The results of performance comparison are given in table 1. In performance comparison, verilog HDL description is used for the MC-NoC and other part of the SoC. Therefore, performance comparison result derived from the simulation is cycle-accurate. In table 1, numbers in “Cycle Count” column means required clock cycles to perform edge detection for 320 x 240 pixels of image. The number in right-most column shows cycle count ratio compared to the task mapping configuration in Fig. 13 (a).

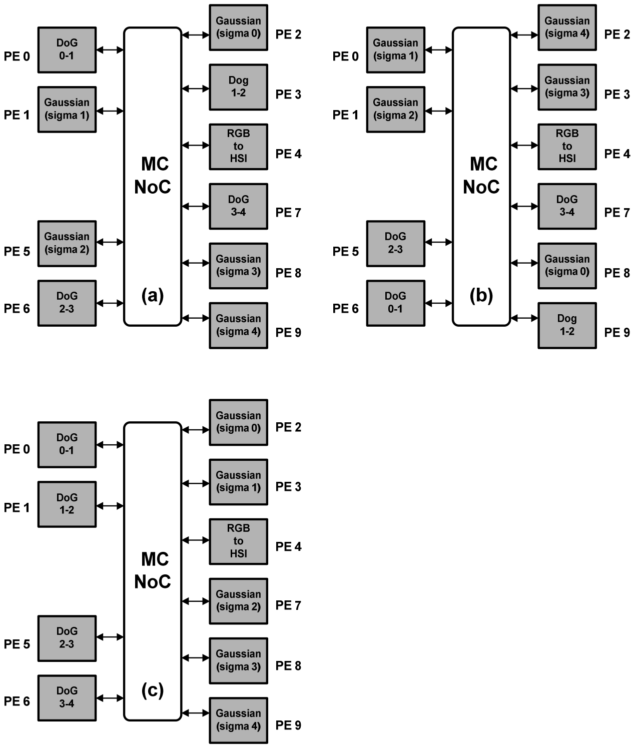


Figure 13. Different Task Mappings of Edge Detection Operation

TABLE I. PERFORMANCE COMPARIONS OF DIFFERENT TASK MAPPINGS ON THE MC-NOC

Task Mapping	Cycle Count	% ratio to mapping (a)
(a)	8,202,900	1
(b)	8,086,580	0.986
(c)	8,021,820	0.978

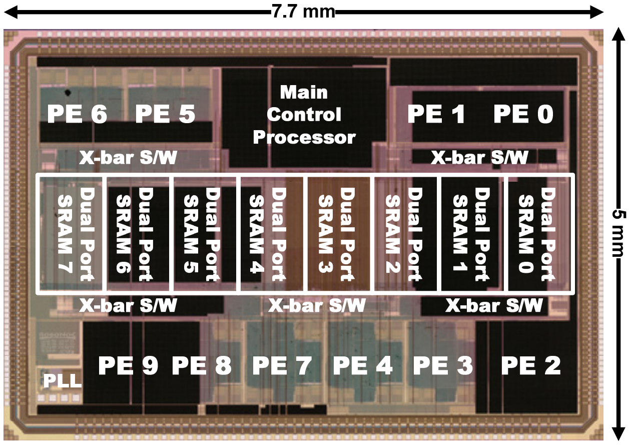


Figure 14. Chip micrograph of the MC-NoC based SoC

The result in table 1 proves the flexibility of the task mapping on the proposed MC-NoC. In the MC-NoC based homogeneous SoC, difference in overall performance according to the various task mapping is less than 3%. This feature of the MC-NoC also facilitates software level optimization of NoC.

VI. Implementation Result

As mentioned in previous section, the MC-NoC and other peripheral parts of the SoC shown in Fig. 9 (a) were designed using verilog HDL. After that verified HDL is synthesized and placed & routed using TSMC 0.18um process library. However, the dual port SRAMs of the MC-NoC is designed using full custom design methodology to reduce area overhead of valid check logic. The chip micrograph of the MC-NoC is shown in Fig. 14. The size of chip is 7.7mm x 5mm and the operation frequency of the MC-NoC is designed to be 400MHz while other part of the SoC operates at 200MHz clock frequency. In the MC-NoC implementation, strobe signal based clock synchronizers, such as introduced in section 3, are also integrated into the chip for clock synchronization between crossbar switch and other part of the SoC.

VII. Conclusion

In this paper, we introduced real chip implementation issues of NoC and provided detailed architectural and circuit level solutions. The presented solutions are based on the previous works performed by authors' researches group. After that, memory centric NoC (MC-NoC) is proposed regarding architectural and circuit level solutions presented in section 3. As a result, hierarchical star topology network and synchronizer design is adopted from the previous design experiences. The proposed MC-NoC is feature with flexibility and feasibility of task mapping on homogeneous SoC. Throughout researches including the proposed MC-NoC, we presented the practical chip implementation of NoC for both heterogeneous and homogeneous SoCs.

References

[1] Dally, W. J., Towles, B., "Route Packets, Not Wires : On-Chip Interconnection Networks", IEEE Proceedings of Design Automation Conference, pp. 684-689, June 2001.

[2] Luca Benini and Giovanni De Micheli, "Networks on Chips : A New SoC Paradigm", IEEE Computer, vol. 35, pp. 70-78, 2002.

[3] Srinivasan Murali and Giovanni De Micheli, "SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs", IEEE Proceedings of Design Automation conference, pp. 914-919, June 2004.

[4] Kees Goossens, John Dielissen, and Andrei Radulescu, " AEthereal Networks on Chip: Concepts, Architectures, and Implementations", IEEE Design & Test of Computers, Vol. 22, Issue 5, pp. 414-421, Sept.-Oct., 2005.

[5] Davide Bertozzi and Luca Benini, "Xpipes : A Network-on-chip Architecture for Gigascale Systems-on-Chip" IEEE Circuits and Systems Magizine, Vol. 4, Issue 5, pp 18-31, 2004.

[6] Jingcao Hu and Radu Marculescu, "DyAD – smart routing for networks-on-chip", IEEE Proceeding of Design Automation conference, pp.260-263, June 2004.

[7] Intel web site - http://www.intel.com/quadcoreserver/

[8] Se-Joong Lee, et. al. "An 800MHz Star-Connected On-Chip Network for Application to Systems on a chip", IEEE Digest of Internaltional Solid State Circuits Conference, vol. 1, pp. 468-489, February, 2003.

[9] Kangmin Lee, et. al. "A 51mW 1.6GHz On-Chip Network for Low-power Heterogeneous SoC Platform", IEEE Digest of International Solid State Circuits Conference, vol. 1, pp.152-518, February, 2004

[10] Se-Joong Lee, et. al. "Adaptive Network-on-Chip with Wave-Front Train Serialization Scheme", IEEE Digest of Symposium on VLSI Circuits, pp. 104-107, June, 2005

[11] Kangmin Lee, Se-Joong Lee, and Hoi-Jun Yoo, "Low-Power Networks-on-Chip for High-Performance SoC Design", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no.2, pp.148-160, February 2006.

[12] Se-Joong Lee, Kangmin Lee and Hoi-Jun Yoo, "Analysis and Implementation of Practical Cost-Effective Network-onChips, IEEE Design & Test of Computers Magazine (Special Issue for NoC), Septempber 2005.

[13] Donghyun Kim, Kangmin Lee, Se-joong Lee and Hoi-Jun Yoo, "A Reconfigurable Crossbar Switch with Adaptvie Bandwidth Control for Netwokr-on-Chip", IEEE International Symposium on Circuits and Systems, Vol.3 pp.2369-2372, May 2005.

[14] Taylor, M. B. et. al., "The Raw Microprocessor : A Computational Fabric For Software Circuits and General-Purpose Programs", IEEE Micro, vol. 22, Issue 2, pp. 25-35, March-April 2002

[15] V. Nolle, et. al., "Operating-system controlled network on chip", IEEE Proceedings of Design Automation Conference, pp.256-259, June 2004.

[16] Se-Joong Lee, et. al. "Packet-Switched On-Chip Interconnection Network for Systems-on-Chip Applications", IEEE Transactiosn on Circuits and Systems II, vol. 52, no. 6, pp. 308-312, June 2005.

[17] C. Svensson. "Optimum voltage swing on on-chip and off-chip interconnect", IEEE Jorunal of Solid-State Circuits, vol. 36, no. 7, pp. 1108-1112, July 2001.

[18] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", ACM International Journal of Computer Vision, Vol. 60, Issue 2, pp. 91-110, 2004