Low-power network-on-chip for high-performance SoC design

Article *in* IEEE Transactions on Very Large Scale Integration (VLSI) Systems · March 2006

DOI: 10.1109/TVLSI.2005.863753 · Source: IEEE Xplore

3 authors, including:

Kangmin Lee
Qualcomm
22 PUBLICATIONS **696** CITATIONS

Low-Power Network-on-Chip for High-Performance SoC Design

Kangmin Lee, *Student Member, IEEE*, Se-Joong Lee, *Member, IEEE*, and Hoi-Jun Yoo, *Senior Member, IEEE*

***Abstract*—An energy-efficient network-on-chip (NoC) is presented for possible application to high-performance system-on-chip (SoC) design. It incorporates heterogeneous intellectual properties (IPs) such as multiple RISCs and SRAMs, a reconfigurable logic array, an off-chip gateway, and a 1.6-GHz phase-locked loop (PLL). Its hierarchically-star-connected on-chip network provides the integrated IPs, which operate at different clock frequencies, with packet-switched serial-communication infrastructure. Various low-power techniques such as low-swing signaling, partially activated crossbar, serial link coding, and clock frequency scaling are devised, and applied to achieve the power-efficient on-chip communications. The 5 $\times$ 5 mm^2 chip containing all the above features is fabricated by 0.18-μm CMOS process and successfully measured and demonstrated on a system evaluation board where multimedia applications run. The fabricated chip can deliver 11.2-GB/s aggregated bandwidth at 1.6-GHz signaling frequency. The chip consumes 160 mW and the on-chip network dissipates less than 51 mW.**

***Index Terms*—Bus coding, crossbar, interconnection, low-power, network-on-chip (NoC), on-chip network, packet, serial communications, small swing, system-on-chip (SoC).**

I. Introduction

SYSTEM-ON-CHIP (SoC) design in the nanoelectronics era brings us not only many opportunities, but also many challenges. More than a billion transistors are expected to be integrated on a single chip in this decade encompassing numerous semiconductor intellectual property (IP) blocks and customized processing units (PUs) [1]. In addition, wire delays have become more critical than gate delays causing performance degradation and synchronization problems between IPs [2]. The synchronization problem due to global wire delays worsens as the clock frequencies increase and the feature sizes decrease. The global interconnections covering the whole chip area are readily influenced by process uncertainty or electrical disturbance which may cause transmission errors. Furthermore, the communication among IPs consumes a significant portion of overall system power budget. Therefore, the performance of SoCs will be determined by how to interconnect efficiently the predefined and preverified IPs as well as how to accommodate their communication requirements.

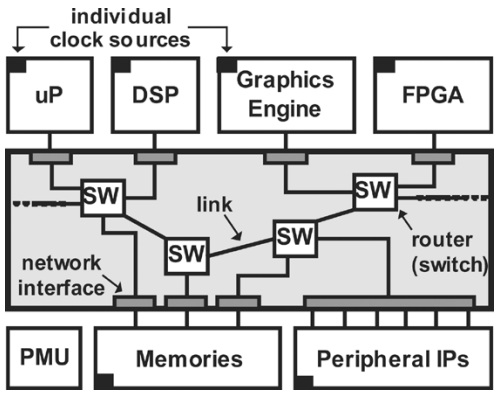


Fig. 1. Heterogeneous NoC architecture.

Recently, network-on-chip (NoC) architectures are emerging as a candidate for the highly scalable, reliable, and modular on-chip communication infrastructure platform [3]. The NoC architecture uses layered protocols and packet-switched networks which consist of on-chip routers, links, and network interfaces on a predefined topology, as illustrated in Fig. 1. There have been many architectural and theoretical studies on NoCs such as design methodology [2], [3], topology exploration [4], quality-of-service (QoS) guarantee [5], resource management by software [6], and test and verifications [7].

In large-scale SoCs, the power consumption on the communication infrastructure should be minimized for reliable, feasible, and cost-efficient implementations. However, little research has reported on energy- and power-efficient NoCs at a circuit or implementation level, since most of previous works have taken a top-down approach and they did not touch the issues on a physical level, still staying in a high-level analysis. Although a few of them were implemented and verified on the silicon [8], [9], they were only focusing on performance and scalability issues rather than the power-efficiency, which is one of the most crucial issues for the practical application to SoC design.

In this work [10], we designed and implemented a heterogeneous NoC focusing on low-energy and low-power communication in various design levels such as circuits, signaling, channel coding, protocol and topology. The fabricated chip emulates a large-scale SoC for an embedded system and contains a variety of IPs such as multiple processors, memories, an off-chip gateway, and peripherals. The integrated on-chip network of a hierarchical star topology provides 11.2 GB/s aggregate bandwidth and consumes 51 mW when the integrated IPs execute load/store operations without idle states, which is the maximum

Manuscript received March 14, 2005; revised August 5, 2005, and October 4, 2005.

K. Lee and H.-J. Yoo are with the Semiconductor System Laboratory, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: kangmin@eeinfo.kaist.ac.kr; kangmin@ssl.kaist.ac.kr; hjyoo@ee.kaist.ac.kr).

S.-J. Lee was with the Semiconductor System Laboratory, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea. He is now with Texas Instruments Incorporated, Dallas, TX 75243 USA (e-mail: shocktop@eeinfo.kaist.ac.kr).

Digital Object Identifier 10.1109/TVLSI.2005.863753

TABLE I
DESIGN PARAMETERS OF THIS IMPLEMENTATION IN 0.18-μm CMOS TECHNOLOGY

Category	Description	Typical value	Symbol
Energy (J)	1-packet write and read	1.97 x 10^{-10}	E_{Queue}
	1-packet switching fabric	6.25 x 10^{-12} × (# of switch ports)	E_{SF}
	1-packet arbitration energy	1.79 x 10^{-13} × (# of switch ports)	E_{ARB}
	1-packet 1mm metal routing	4.38 x 10^{-11}	E_{Link}
	1-packet 1mm metal routing (P-tc-P)*	8.76 x 10^{-11}	E_{Link_PtP}
Area (μm^2)	3-packet queuing buffer	8.40 x 10^4	A_{Queue}
	Crossbar switching fabric	1.47 x 10^3 × (# of switch ports) *	A_{SF}
	Arbitration logic	2.70 x 10^3 × (# of switch ports)	A_{ARB}
	20b 1mm metal routing (FWD+BWD)	3.80 x 10^4	A_{Link}

* The point-to-point topology consumes much more metal routing resources than other topologies do. Therefore upper metal layers should be fully used. This situation increases the wire metal coupling capacitance vertically, thus the link energy consumption also increases.

traffic condition in the system. The ratio of power consumption to sustained bandwidth is reduced by ten times from that of the previous work [8] by means of various low-power techniques which will be presented later in this paper.

The organization of this paper is as follows. In Section II, we describe the NoC architecture from a viewpoint of power-efficiency. Section III presents the low-power techniques used in this study in detail. Section IV reports the implementation and measured results of the fabricated NoC. Finally, the paper concludes with Section V.

II. NoC Architecture

A. Hierarchical Star Topology

The first step for NoC architecture design is choosing an optimal NoC topology for the target system. In this chapter, popular and interesting topologies are compared briefly in practical terms of energy consumption and area cost. There are basic topologies such as a bus, star, mesh, and point-to-point topology and also hierarchical topologies which could have the same or different topologies locally and globally, for example, a locally bus globally mesh topology. We use an average packet traversal energy $E_{\rm pkt}$ as a network energy efficiency metric [11] which can be estimated by the following equation [2], summing up the energies on switching hops, links and a destination queue:

$$E_{\rm pkt} = H_{\rm Avg} \cdot (E_{\rm Queue} + E_{\rm SF} + E_{\rm ARB}) + L_{\rm Avg} \cdot E_{\rm Link} + E_{\rm Queue} \quad (1)$$

where $H_{\rm Avg}$ and $L_{\rm Avg}$ are average hop counts and an average distance, respectively, between a sender PU and a receiver PU. Energy consumption on a switching hop is composed of energy consumption in an input queuing buffer or latch $E_{\rm Queue}$, switching fabric $E_{\rm SF}$, and arbitration logic $E_{\rm ARB}$. $E_{\rm Link}$ stands for transmission energy on a unit length link. Those energy terms are measured from the circuit implementation result in 0.18-μm technology (see Table I).

The energy and area cost analyses are performed with various numbers of PUs (N) from 16 to 100. The hierarchical topologies are assumed to be divided into $\sqrt{N}$ of clusters and each cluster contains $\sqrt{N}$ PUs. For the simplicity of the calculation, the sizes of PUs are assumed as the same size as 1 mm $\times$ 1 mm and the PUs are placed as a square matrix regardless of the topology.[1] Mesh and star networks perform input buffered packet switching at every switching node, while bus and point-to-point networks do not have internal buffer in their network. There are two kinds of traffic pattern; one is uniform random traffic and the other is localized traffic with a locality factor. The locality factor means a ratio of the intracluster traffic to the overall traffic. It is obvious that PUs requiring low latency and large bandwidth communication can get more synergetic performance by locating them in the same cluster based on their communication locality so that the intracluster traffic is dominant over all of on-chip traffic. The locality factor can represent the localized traffic pattern quantitatively.

Fig. 2(a) shows the comparison result of the energy consumption under the uniform traffic. Among the flat topologies, dashed lines in the figure, the energy costs are decreasing in the order of "bus > mesh> star> point-to-point." Although the mesh has short and regular length of links, it has more hop counts than the star thus the energy cost of the mesh is 40%–50% higher than the star. Among the hierarchical topologies excluding the hierarchical point-to-point topology, the hierarchical star (locally star globally star or H-star) topology shows the lowest energy cost

[1] In the actual fabricated chip, the integrated IPs such as microprocessors and SRAMs occupy the similar size as 1 mm × 1 mm like our assumption.

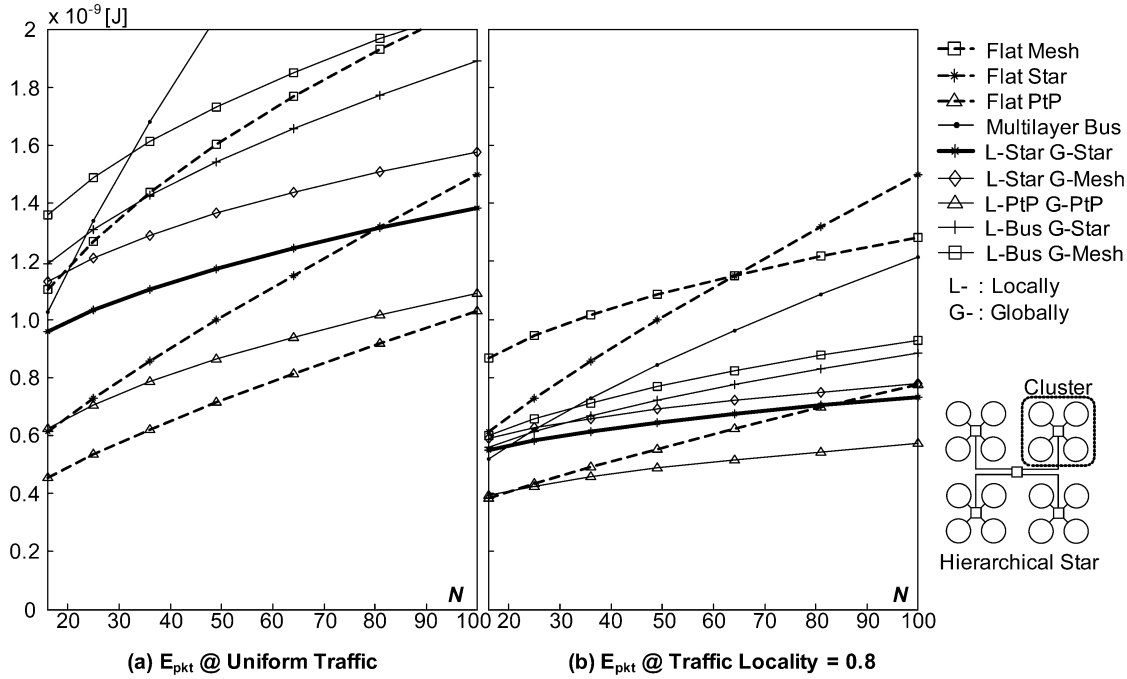


Fig. 2. Energy consumption according to a number of PUs.

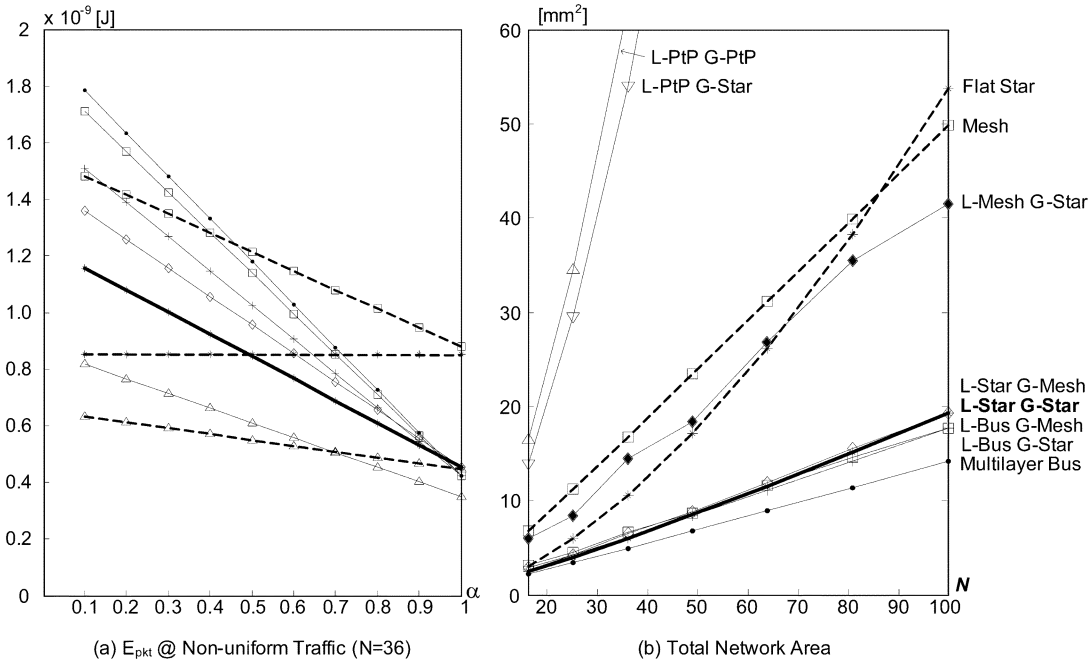


Fig. 3. (a) Energy consumption according to the traffic locality factor ($N = 36$) and (b) network area according to a number of PUs.

under any kinds of traffic. Fig. 3(a) shows the energy cost according to the traffic locality when N is 36 as an example.

The network area cost including the area of switches, multiplexers/demultiplexers, and links are also analyzed as shown in Fig. 3(b). The area of point-to-point topologies, triangle symbols in the figure, is skyrocketing as the N increases because of their huge link wires interconnecting every PU pair. This is the major reason which makes the point-to-point topology impractical to implement. The area costs of a flat mesh and a flat star topologies increase rapidly due to the linearly increasing buffer area of the mesh and the superlinearly increasing switch fabric area of the star. The flat mesh or the flat star network occupies almost the 30% of the entire chip area. The area consumption of the hierarchical topologies is as small as bus topologies. The hierarchical star network occupies only the 10%–15% of the overall chip area. Considering the energy and area cost together, the hierarchical star topology is the most energy-efficient and cost-effective topology in general.

B. Hierarchical Circuit and Packet Switching

In the proposed H-star topology, the local intracluster network and the global intercluster network have different traffic properties and requirements so that different switching methods should be applied to meet the requirement of each network. In

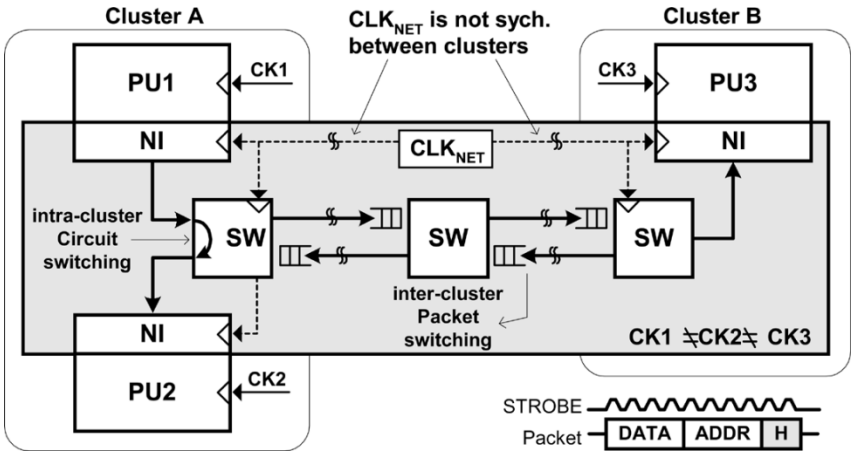


Fig. 4. Synchronization structure in the NoC.

the intracluster network where the most traffic is between processors and local memories, network latency should be more emphasized than throughput since the predictable and deterministic latency is a crucial requirement for real-time applications or for the simplicity of software programming. Thus, we apply circuit switching to the local intracluster network where a physical path from the source to the destination is reserved prior to the transmission of the data. Once a transmission starts, the transmission is not corrupted by other transmission and packets are not stored in buffer; thus, deterministic packet delay is guaranteed. Moreover, the circuit switching does not need packet buffers so that area and power consumption can be reduced in the implementation of intracluster networks. The overhead in latency which is necessary for forming and removing the requested circuit is minimized in this topology since the intracluster network is a star-topology which has a single hop count.

On the other hand, the global intercluster traffic shares the bandwidth of the switch-to-switch link; thus, the throughput of the shared and limited link is more important rather than the latency. Packet switching permits statistical multiplexing on the shared cluster-to-cluster channel. That is, the packets from many different sources can share the same channel for an efficient use of the fixed capacity like Internet on computer communications. Therefore, the packet switching is appropriate for global intercluster network. Packet buffers should be placed at the both ends of the intercluster channel, costing extra area and power. The shift-register type buffer of single packet capacity takes 200×140 μm area and consumes 6.5 mW at 1.6-GHz signaling frequency.

C. Synchronization

A state-of-the-art SoC is a heterogeneous multiprocessing system with multiple timing references, because of the difficulty of global synchronization as well as PU-independent power management like dynamic frequency scaling. Therefore, synchronization between subsystems located in different clock domains becomes more crucial.

Fig. 4 shows the proposed synchronization structure in such a SoC with multiple clock domains. Although each processing unit operates with its own clock, CK_n, it uses a unique clock, $\mathrm{CLK_{NET}}$, for the purpose of the communication with other PUs.

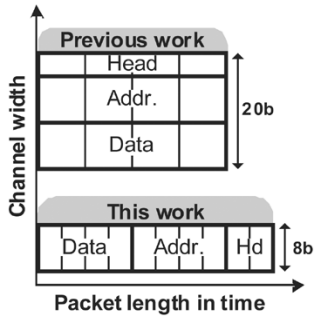


Fig. 5. Two serialization methods.

Network interface (NI) translates the timing reference of CK_n to that of $\mathrm{CLK_{NET}}$ and vice versa for the packet transmission. It is possible to synchronize the $\mathrm{CLK_{NET}}$ inside a cluster because the physical area of a cluster can be covered by a single clock domain. But, it is usually difficult to synchronize the $\mathrm{CLK_{NET}}$ for all clusters in the SoC. Therefore, intercluster communications become mesochronous—the same frequency but with different phase—and an additional synchronization scheme is needed for the reliable global transmission of data. In this implementation, the source-synchronous scheme is adopted. That is, a strobe signal is transmitted along with packet data for the timing reference at a receiver end and packet buffer between switches plays the role of the first-in first-out (FIFO)-synchronization. Since the unified clock, $\mathrm{CLK_{NET}}$, need not be synchronized between clusters, the simultaneous switching noise on the power supply can be reduced, and higher power/ground integrity and less electromigration effect can be obtained.

D. On-Chip Serialization

On-chip serial communication has many advantages over multibit parallel communication in respects of signal skew, crosstalk, area cost, and wiring congestion [8]. In the proposed architectures, a packet of maximum 80 bits composed of 16-bit header, 32-bit address, and 32-bit data is serialized into 8 bits by SERDES circuits inside of the NI. The serialization method is similar to that of the previous work [8] but modified as shown in Fig. 5. In the previous implementation, header, address, and data have their own channels of 4, 8, and 8 bits, respectively,

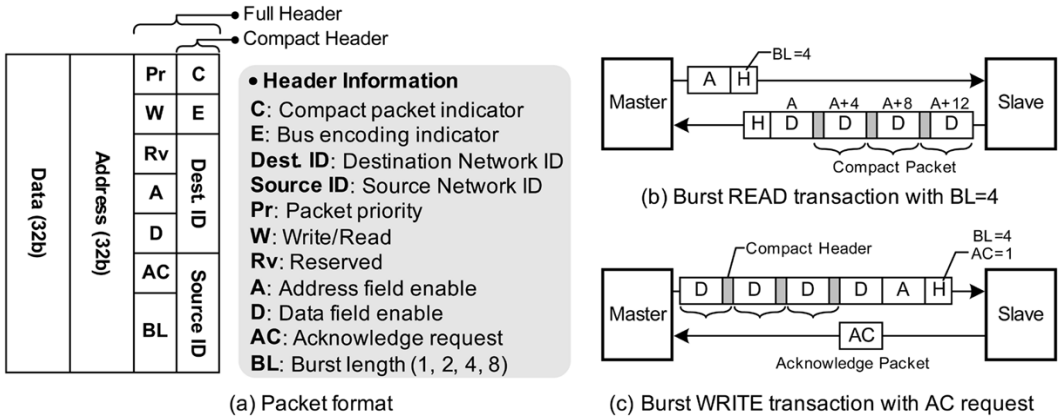


Fig. 6. NoC protocol: (a) Packet format; (b) burst READ transaction; and (c) burst WRITE transaction.

with 800-MHz signal frequency. In this implementation, however, they are multiplexed onto an 8-bit channel in time-sharing with 1.6-GHz frequency. Therefore, the area of a switch fabric is further reduced to 1/4 of the previous one by narrowing the channel width. Regarding the network bandwidth, our serialization style has an advantage over the previous one because the packet structure is simpler. For example, in case of the READ operation, the request and response packets need not contain data field and address field, respectively. While the previous serialization method always had the fixed packet length, the proposed architecture can have shorter packet length by removing the unnecessary field according to the protocol which will be described in Section II-E. As a result, the utilization of the shared channel can be increased.

E. NoC Protocol

Fig. 6 shows the Basic On-chip Network (BONE) protocol used in packet transactions [12]. The NoC protocol supports burst packet transactions for large data transmissions with length of 2, 4, and 8 packets. Burst READ request packet contains only the base address but its response packet contains successive read data starting from the base address to the address increased by the burst-length as shown in Fig. 6(b).

When there is a burst transaction between a faster master and a slower slave, the channel may be under-utilized during the burst transactions. To avoid the throughput degradation in this implementation, the channel can be shared with other packet flows when the channel is idle, even during burst transactions. In that case, each of the burst beat should have its routing-information in its header, which is a compact header. The first READ response packet has full information but remaining packets have the minimum information required for routing in their compact headers. Burst WRITE request of length 4 sends only the base address in the first packet as shown in Fig. 6(c). The following packets have compact header and data fields. By using the burst transaction with compact packets, total transaction time can be reduced by half compared with multiple single-packet transactions.

In the implemented protocol, the packet format has 3 bits of source ID and 3 bits of destination ID; therefore, it supports 8 masters and 8 slaves in maximum. To scale up the network size, you should increase the ID fields in the packet format before the chip design. A 1-bit of priority information in header field is asserted by software on each PU for the urgent transaction and it enables the differentiated scheduling among ordinary packets. For more reliable transaction, the handshaking is supported by using an acknowledgment request as shown in Fig. 6(c). The NoC protocol provides a 1-bit sideband back-pressure signal for congestion control in the networks. The back-pressure signal is asserted when a packet buffer exceeds a predetermined threshold, or when the destined PU cannot provide service temporarily.

III. Low-Power Techniques

A. Low-Swing Signaling

The global link that connects two clusters is usually a few millimeters long in a large SoC and consumes higher power than a local link does. Low-swing signaling can alleviate its energy consumption significantly [13]. Fig. 7 shows the differential low-swing signaling scheme and its transceiver circuits used in this implementation. The 5.2-mm global wires without repeaters are laid out in zigzag shape to emulate the interconnection on a large chip. A transmitter drives the wires using V_{SWING} which is smaller than V_{DD}, and a receiver restores the signal swing to its full-swing voltage, V_{DD}. The driver uses n-MOSFET for both pull-up and pull-down transistors instead of a CMOS inverter to exploit their lower linear resistance at small drain–source voltage. A simple clocked sense amplifier and a three-stage inverter chain amplify the low-swing signal to the full logic level. PMOS are used as the input gates in order to receive a low common-mode input signal. The sizes of the input gates and their bias currents are chosen to amplify the differential input of as low as 200-mV swing to 1.6-V full-logic swing with small delay. A clock signal for the clocked sense amplifier is regenerated from a strobe (STB) signal by a clock restore circuits (CRC). In the CRC, an inverted strobe input, NSTB, is fed to the P1-gate in order to reduce the standby power consumption. When there is no packet transmission, nSTB goes higher to turn off the CRC bias current.

There was a report on the existence of an optimum voltage swing on long wire signaling for the lowest energy dissipation [14]. To find out the optimum voltage swing at which the energy and delay product has the smallest value, we conducted

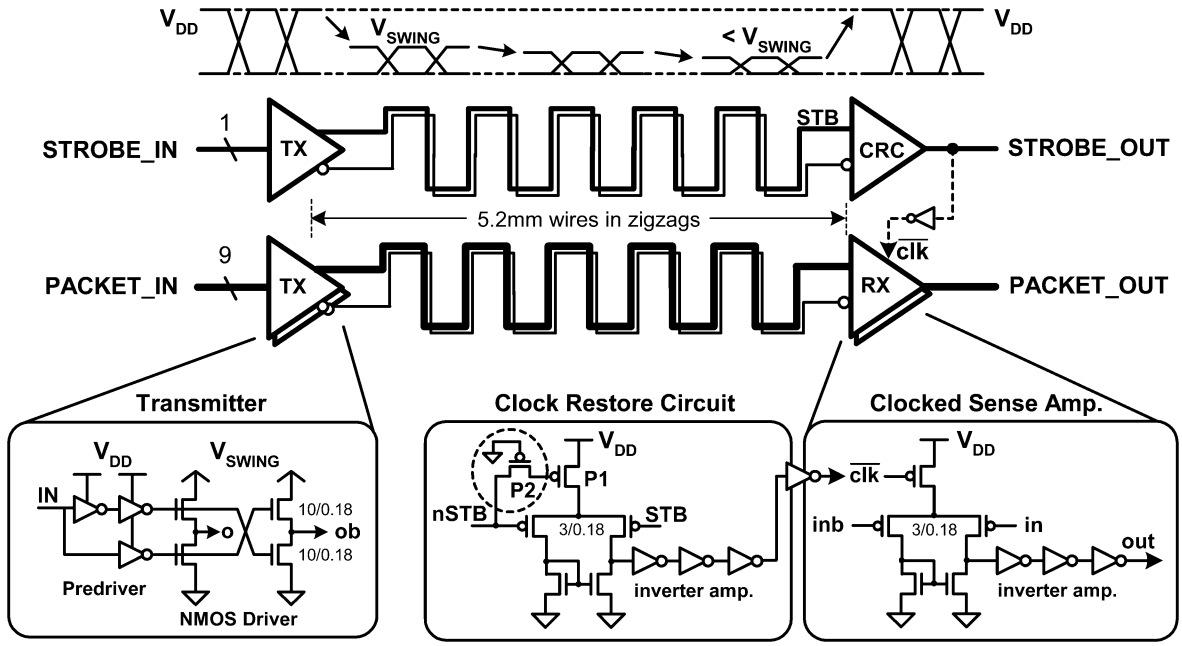


Fig. 7. Low-swing signaling and its transceiver circuits.

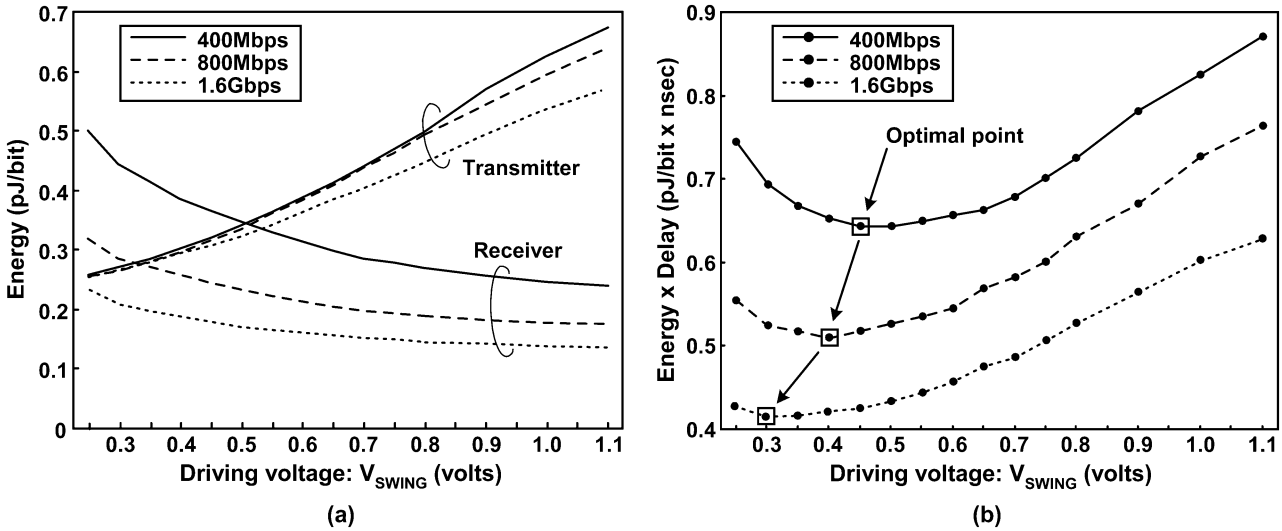


Fig. 8. (a) Energy consumption. (b) Energy and delay product versus voltage swing at various signaling rates.

the post-layout simulations with premeasured capacitance and resistance values. A 5.2-mm metal2 wire of 0.5-μm width and 1.1-μm space has 330-fF parasitic and 100-fF coupling capacitance values. V_{SWING} scans from 0.25 to 1.1 V with 50-mV step when signaling rates are 400 Mb/s, 800 Mb/s, and 1.6 Gb/s as shown Fig. 8(a). The required energy on the transmitter to create a certain voltage swing on the wires decreases linearly with the swing level, whereas the energy to amplify this signal back to its normal logical swing level increases superlinearly with the decrease of V_{SWING} level. The optimum V_{SWING} exists due to such opposite trends of the required energy in the transmitter and the receiver.

Fig. 8(b) shows energy and delay product versus V_{SWING}. The delay from a transmitter to a receiver is about 0.9 ns and its variation is as small as ± 40 ps. As shown in the figure, the optimal swing voltage is 0.45, 0.40, and 0.30 V at 400 Mb/s, 800 Mb/s, and 1.6 Gb/s signal rates, respectively. (Based on the measurement results, the low-voltage goes down to 0.27 V without transmission error.) At each signaling rate, the driving voltage scales to the optimal voltage which is obtained in the previous analysis. Due to the low-swing signaling, the power dissipation on the global link is reduced to 1/3 of that on a full-swing repeated link and no repeaters are used on the wires to avoid area overhead.

B. Mux-Tree Based Round-Robin Scheduler

A scheduler (or arbiter) is needed in a crossbar switch when more than two input packets from different input ports are destined for the same output port at the same time. Among a number of scheduling algorithms, a round-robin algorithm is most widely used in asynchronous transfer mode (ATM)

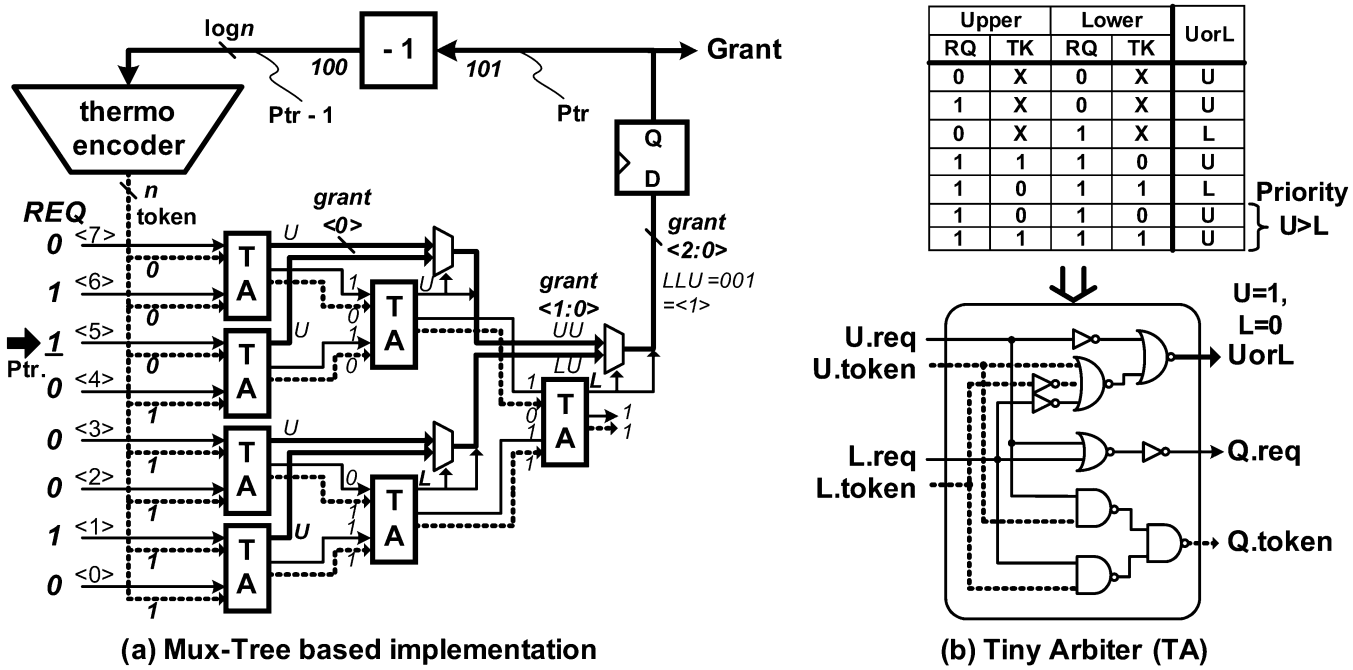


Fig. 9. Mux-tree-based round-robin scheduler. (a) Block diagram. (b) Tiny arbiter.

switches and on-chip networks due to its fairness and lightness [15]. There are many ways on how to implement the round-robin algorithm [15], [16]. In this paper, we propose a new Mux-Tree based implementation for high modularity and scalability as shown in Fig. 9. Its scheduling latency is $O(\log n)$ and required resources are $O(n)$, where n is the number of input ports in a crossbar switch.

The round-robin scheduler has a rotating pointer that indicates the most recently granted port. A port next to the pointer has the highest priority to be granted. For example, request vector $\langle 7{:}0\rangle = 01\underline{1}00010$ is shown in Fig. 9(a) where underline means a current position of the pointer. Then, port$\langle 4\rangle$ has the highest priority and the lower group of port$\langle 4{:}0\rangle$ has higher priority than upper group of port$\langle 7{:}5\rangle$. This information is given by a thermo-encoder whose output becomes token $\langle 7{:}0\rangle = 00011111$. Therefore, port$\langle 4{:}0\rangle$ have their tokens while port$\langle 7{:}5\rangle$ do not. A request from a port having a token acquires higher priority than others. These request and token vectors are inputs of the binary Mux-Tree which is composed of tiny arbiters (TA) at each node. Each TA selects one of two ports, upper one or lower one, based on a table shown in Fig. 9(b). When both of the two requests have no token or both of them have their tokens, TA selects upper port because the pointer rotates in decreasing order. Then, the TA forward the winners request and token to its parent node. Then one of two childrens *UorL* bits is selected by 2:1 MUX based on their parents *UorL* bit. The selected child-*UorL* bit and its parent-*UorL* bit are concatenated and propagate to its grandparent node. By the successive propagation up to the root node, the granted port number, $grant\langle 2{:}0\rangle$, is determined finally.

The proposed Mux-Tree based implementation is compared with four other designs such as EXH, SHFT_ENC, RIPPLE, and DUAL_SPE presented in [15]. Fig. 10 shows comparison results of power consumption and scheduling delay simulated

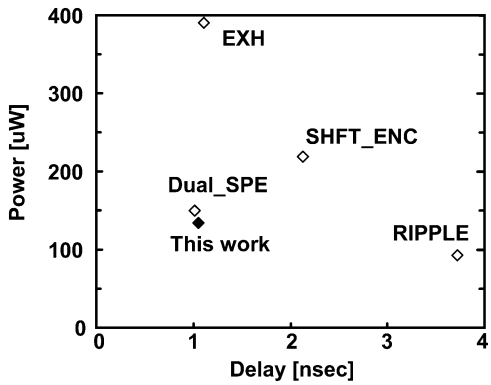


Fig. 10. Power and delay comparison with other round-robin implementations [15].

TABLE II
COMPARISON OF THE NUMBER OF REQUIRED TRANSISTORS

	8 ports	16 ports
RIPPLE	403	927
EXH	1435	6879
SHFT_ENC	629	1711
DUAL_SPE	573	1483
This work	569	1203

with 8-input ports in 0.18-μm process technology. The proposed implementation, Mux-Tree, performs the minimum power and delay product; 136 μW and 1.05-ns delay at 100-MHz clock frequency with offered load of 50%. The proposed scheduler requires the minimum number of transistors, i.e., silicon area, except the RIPPLE design as shown in Table II.

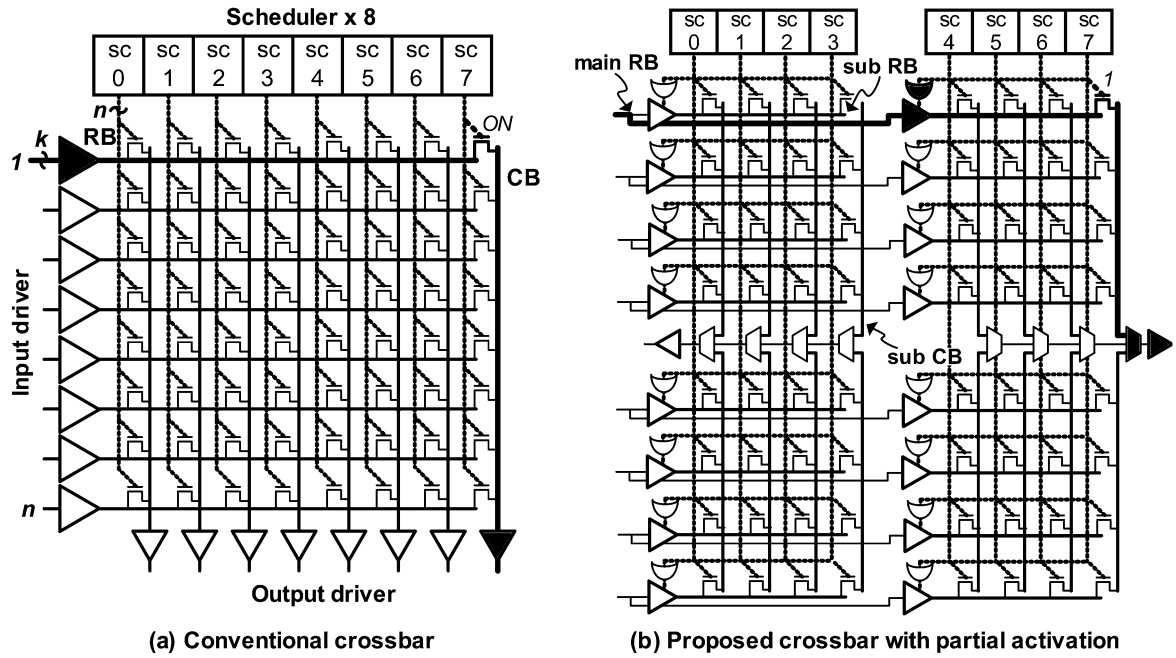


Fig. 11. Schematic diagrams of: (a) an 8×8 conventional crossbar; and (b) a proposed crossbar with partial activation technique.

C. Crossbar Partial Activation Technique

A conventional $n \times n$ crossbar fabric comprises n^2 crossing junctions which contain NMOS pass-transistors as shown in Fig. 11(a). Each input driver wastes its power to charge and discharge two long wires—row-bar (RB) and column-bar (CB)—and $2n$ transistor–junction capacitors. The RB and CB should be laid out with lower metal layers, M1 or M2, in order to reduce the fabric area and to minimize the number of resistive vias. Therefore, the loading on the driver becomes significant as the number of ports increases.

In order to reduce the power consumption, we proposed a crossbar switch with crossbar partial activation technique (CPAT) as illustrated in Fig. 11(b) [10]. By splitting the $n \times n$ fabric into 4×4 fabrics (or tiles), the activated capacitive loading is reduce by $n/4$. A gated input driver at each tile activates its subRB only when the tile gets a grant from its scheduler. For the implementation of CPAT, only 4 four-input OR-gates are needed additionally in each tile. The output line, CB, is also divided into two sub-CBs to prevent the signal propagation into other tiles. A 2:1 MUX connects one of two sub-CBs to the output port based on the grant signals from its scheduler.

An 8×8 crossbar fabric with CPAT is analyzed in comparison with the conventional scheme. The area of the fabric is about $240 \times 240\ \mu\mathrm{m}^2$. According to the capacitance extraction from the layout, the parasitic capacitance values of the RB and the CB are 44 and 28 fF, respectively, and coupling capacitance between adjacent bars is 13 fF in a conventional fabric. Fig. 12 shows the power comparison as a function of the offered loads. At 90% offered load, 22% power saving is obtained. The additional OR-gates and MUXs consume less than 2% of overall power of the crossbar fabric. When CPAT is applied to 16×16 crossbar switch which is dived into sixteen tiles, 43% power can be saved.

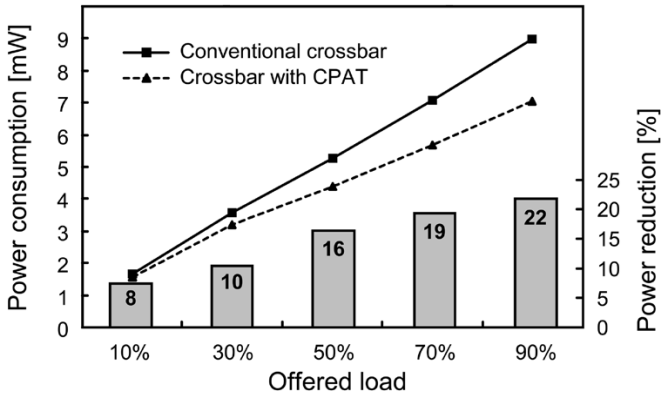


Fig. 12. Power comparison of an 8×8 crossbar fabric with and without the crossbar partial activation technique.

D. Low-Energy Coding on On-Chip Serial Link

In serial communications, the switching activity factor of a serial wire is different from that of parallel wires. The difference in activity factor strongly depends on the transacted data patterns. Fig. 13(a) and (c) shows the comparison results of the number of transitions in parallel with serial communications. In this example, the 8-bit parallel bus has seven transitions. However, when the parallel data are serialized into a serial stream, the number of signal transitions on the wire increase up to 31. The main reason of the increase is the loss of the data correlation between successive data words during serialization. In common multimedia applications, the most significant bits tend to have high correlations because of the sign extension or the locality characteristics of multimedia contents [17]. In these applications, the serial communication dissipates more energy than the parallel communication does.

Many parallel bus coding methods have been proposed to reduce the switching power on the address or data bus connecting

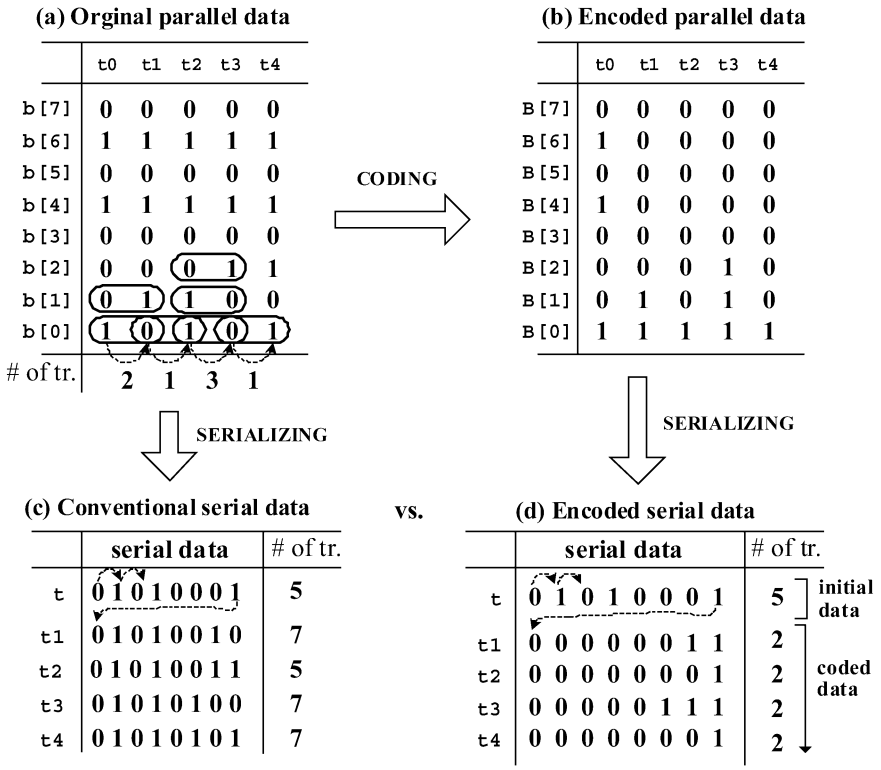


Fig. 13. (a) Original data words with seven transitions; (b) encoded data words; (c) conventional serial data with 31 transitions; and (d) encoded serial data with 13 transitions.

a processor with memories. However, such conventional parallel bus coding methods cannot be employed in the serial bus. Therefore, we proposed a serialized low-energy transmission (SILENT) coding technique [18] to minimize the transmission energy on the serial wire by using the data correlation properties which might be lost during serialization.

In this coding, only the differences between successive parallel data words are encoded as 1's. The encoding algorithm is expressed as follows:

$$\mathbf{B}^{(t)}[i] = \mathbf{b}^{(t)}[i] \oplus \mathbf{b}^{(t-1)}[i], \quad \text{for } i = 0 - (n-1) \qquad (2)$$

$\mathbf{b}^{(t)}[n-1:0]$ n-bit original data word at time t
$B^{(t)}[n-1:0]$ n-bit encoded data word at time t.

By serializing the encoded data words, the frequency of the zero-occurrences on the wire increases because of the correlation of $b^{(t)}$. In Fig. 13(b), all bits from $\mathbf{B}[7]$ to $\mathbf{B}[3]$ become zeros after the encoding because those bits did not change with time. Serializing these encoded words reduces the number of transitions of the serial wire as shown in Fig. 13(d) and the wire looks quiet or even *silent*. In this example, a conventional serial wire without the SILENT coding has three times as many transitions. By reducing the number of transitions on the serial wire, the transmission energy can be saved proportionally. Of course, there is a concern that if a certain packet contains an error, it may be accumulated to the following packets in this coding scheme. In order to prevent such a long error-propagation problem, nonencoded packets can be inserted periodically and the period is even programmable by a sender PU. Therefore, the user can select the optimum period by trading off the error penalty and the power consumption.

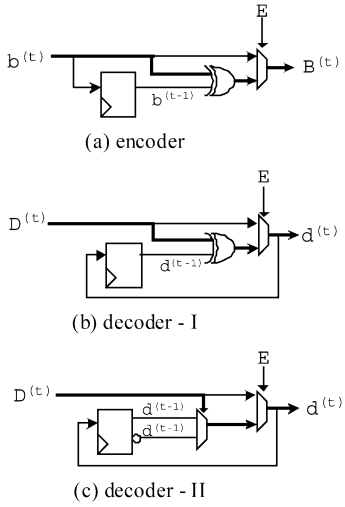


Fig. 14. Circuits implementation of: (a) encoder; (b), (c) decoders.

Fig. 14 shows the circuit implementation of the SILENT CODEC and the bold line indicates a critical path in each circuits. It requires a single XOR delay (less than 100 ps) for encoding or decoding, which is very negligible delay, i.e., 0.01 cycle when clock frequency is 100 MHz. Therefore, there is no throughput degradation caused by the SILENT coding. The power consumption dissipated for 32-bit data word encoding and decoding is about 390 and 385 μW, respectively, at 100-MHz frequency in the worst case data input patterns. This is also negligible power overhead compared with the data transmission power.

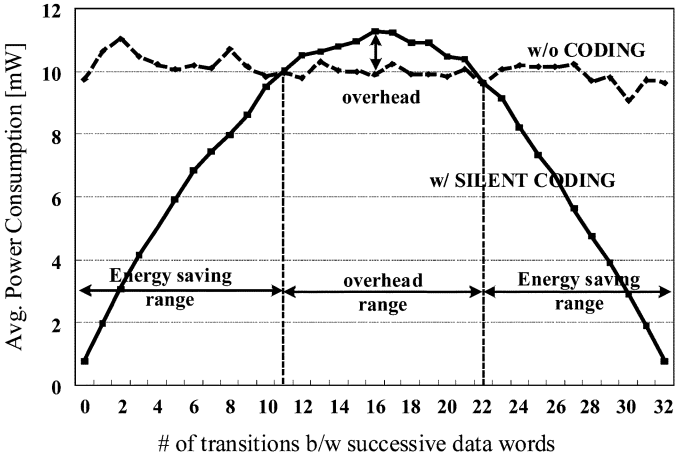


Fig. 15. Average power consumption on serial communications with and without SILENT coding.

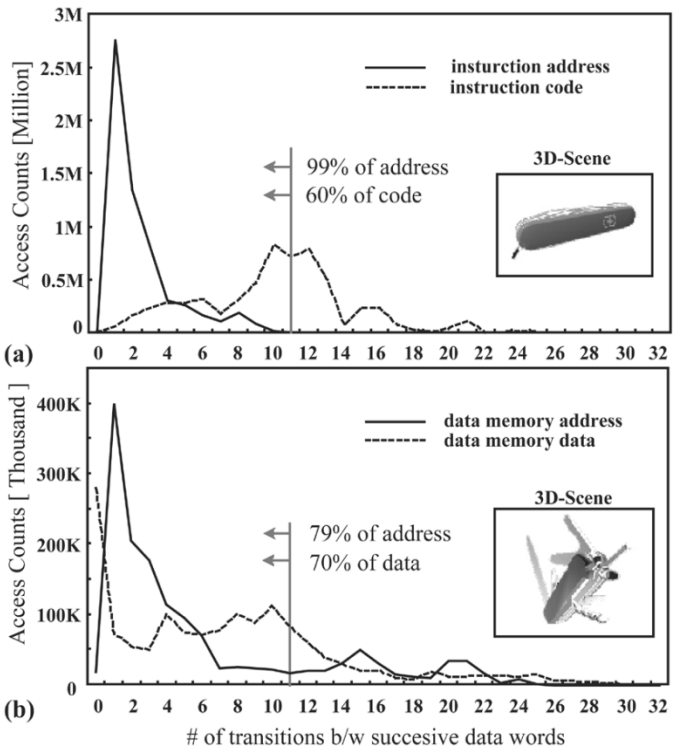


Fig. 16. Distribution of the displacement between successive: (a) instruction accesses; (b) data memory accesses.

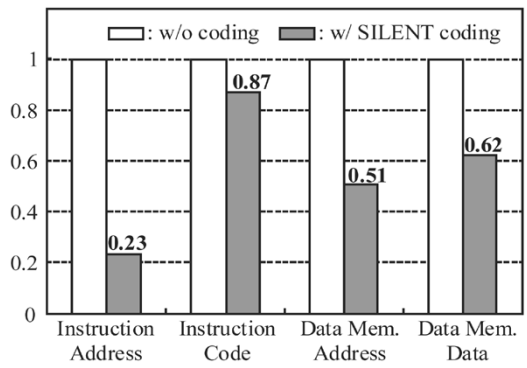


Fig. 17. Normalized average energy consumption in each memory access type.

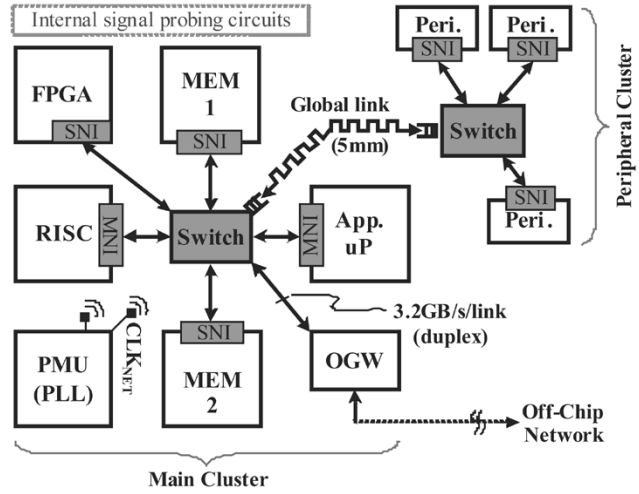


Fig. 18. Block diagram of a prototype SoC for multimedia applications.

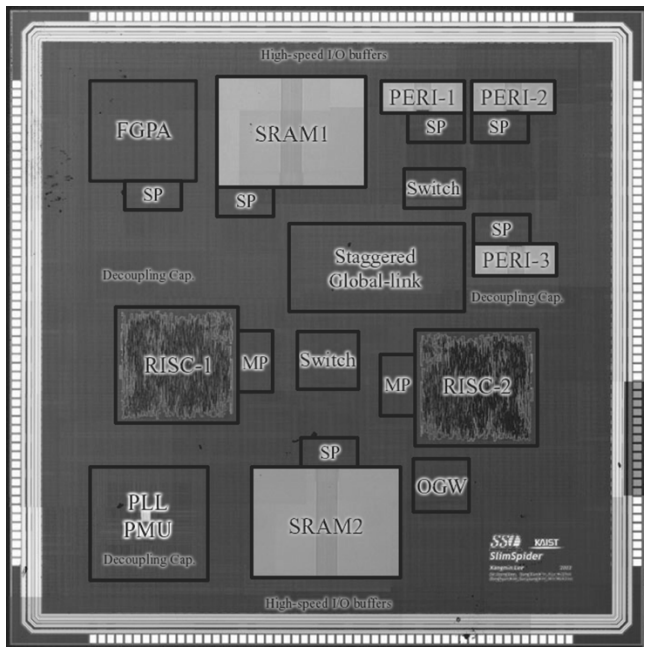


Fig. 19. Die photograph.

In order to analyze the energy efficiency of this coding scheme, we evaluate the energy consumption in the serial communication channel containing 32-bit CODEC circuits, 32-bit-to-4 bit SERDES circuits, and 4-bit 8-mm serial wires. The energy consumption depends on the data patterns required to transmit. All possible data transitions from a random data word are analyzed. Fig. 15 shows the average power consumption with and without SILENT coding scheme at 100-MHz operating frequency. It contains all the power dissipated in links, encoder, decoder, and SERDES circuits. The x-axis stands for the number of data displacement between successive 32-bit data words. The analysis results show that the energy can be saved in the regions under 12 or above 21 in the x-axis by using the SILENT coding. However, power overhead exists at most 14% in a region from 12 to 21. The energy saving range is twice wider than the overhead range and the amount of power saving is much larger than the overhead. Therefore, the

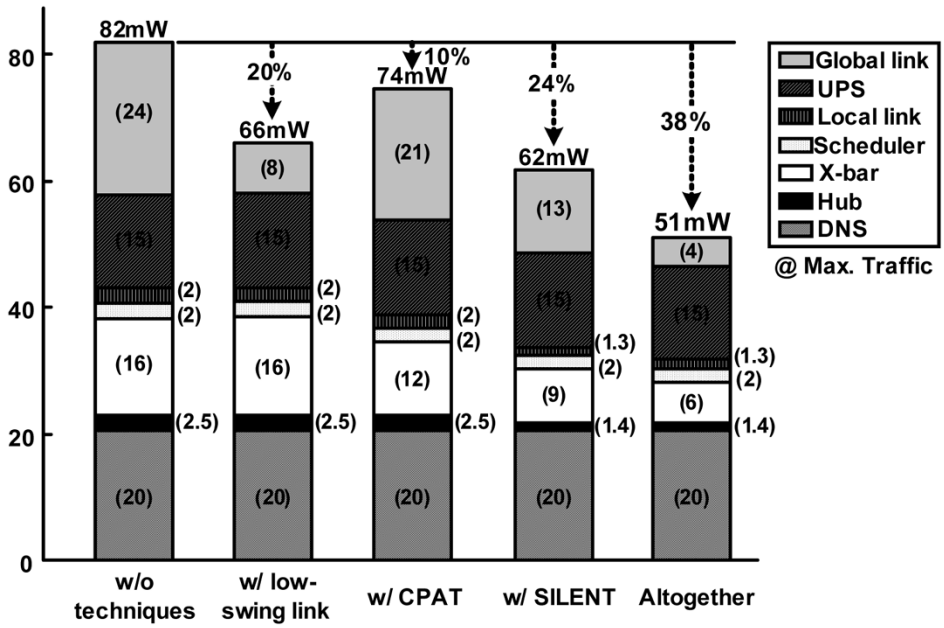


Fig. 20. On-chip network power consumption reduction and breakdown with and without low-power techniques.

SILENT coding has more opportunity to save energy in most of data patterns.

According to [20], in order to save 20% power consumption on a parallel bus by using parallel bus encoding techniques such as Bus-Invert coding [21], the bus should be as long as 15 mm at least to hide the power overhead of CODEC circuits. This seems so long as a chip interconnection between PUs. Thus those parallel bus encoding schemes are addressed difficult to get its power saving effect [20]. Meanwhile, the SILENT coding is proposed for a serial bus not for the parallel bus. The serial bus consumes much more power than a parallel bus does because of the reason described above. It is large enough to hide the power overhead caused by the SILENT CODEC. Moreover, the power-saving efficiency of SILENT is much larger than the conventional parallel bus coding schemes.

For more realistic evaluations of SILENT coding, we trace the transactions of the on-chip traffic between a RISC processor and system memories while a 3-Dimensional Graphics application is running [19]. Full 3-D Graphics pipelines of geometry and rendering operations are executed for 3-D scenes with 5878 triangles. Fig. 16 shows the traffic correlations traced from the memory accesses. The instruction memory address is so sequential that the 99.5% of six million transactions are within the energy saving region of the SILENT coding. Although the instruction codes are quite random pattern, the 60% of it is within the energy saving region. In the case of the data memory access, the 79% and 70% of 1.5 million data memory address and data transactions are within the energy saving region, respectively. With this memory access pattern, the energy consumption is evaluated in the serial communications. Fig. 17 shows the normalized average energy consumption on the serial wire with and without SILENT coding. The SILENT coding shows the best performance for instruction address among the memory access types, about 77% energy saving. Even in the random traffic, in the case of the instruction codes, 13% energy saving is achieved.

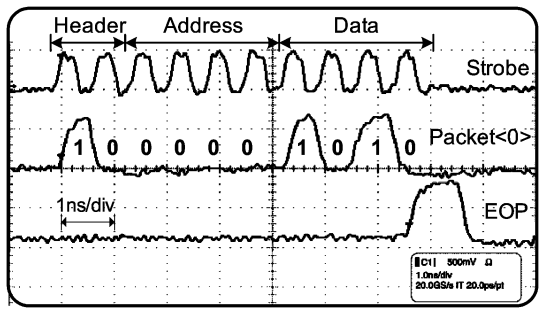


Fig. 21. Measured on-chip packet signals on the network; strobe, $packet\langle 0\rangle$ and EOP.

It also saves 40%–50% of the transmission energy for multimedia data traffic. The analysis shows that the SILENT coding reduces the energy consumption of the serial communication in all kinds of on-chip traffic for the 3-D Graphics application.

IV. Implementation and Measurement Results

We implemented a multimedia SoC as a prototype application by utilizing the proposed NoC architecture, protocol and low-power techniques. The overall block diagram is shown in Fig. 18.

The chip integrates two clusters; a main cluster and a peripheral cluster. The main cluster contains two RISC processors for a multiprocessor system emulation, on-chip FPGA, two 64-kb SRAMs, and an off-chip gateway [8] for seamless off-chip communications with other NoCs. The peripheral cluster contains three 4-kb SRAMs to emulate slow peripheral slaves. We assume that the peripheral cluster is located far from the main cluster to emulate a large SoC, thus two clusters are interconnected by a 5.2-mm low-swing global link. The on-chip PLL generates a 100-MHz clock for main cluster PUs, a 50-MHz clock for peripheral cluster units, and

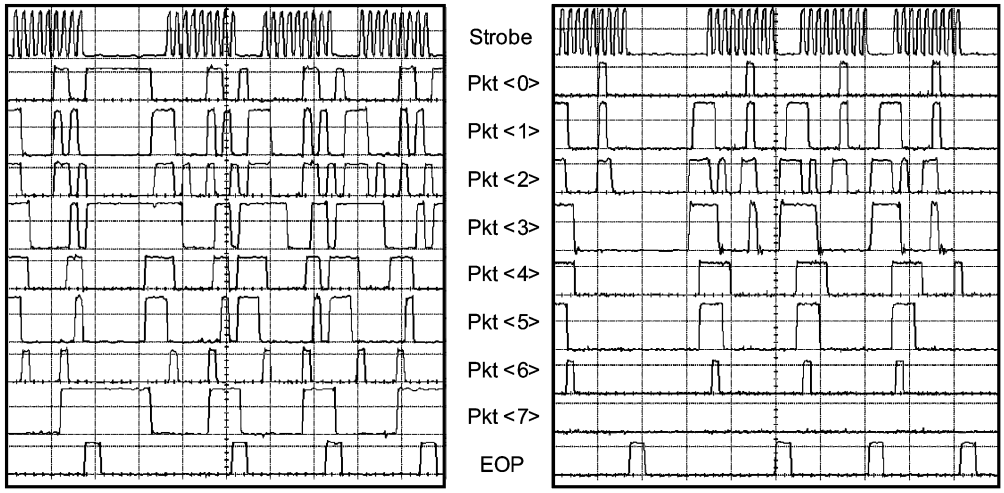


Fig. 22. Measured packet signals: (a) without and (b) with the SILENT coding.

a 1.6-GHz $\mathrm{CLK_{NET}}$ for networks and interfaces. The clock frequencies are scalable in accordance with power management modes; 100/50/1600 MHz for FAST mode, 50/25/800 MHz for NORMAL mode, and 25/12.5/400 MHz for SLOW mode. Each PU clock is not synchronized with each other intentionally for the emulation of large systems which have multiple clock domains.

The on-chip network provides 3.2-GB/s communication bandwidth for each PU and 11.2-GB/s aggregate bandwidth (1.6 GHz $\times$ 8 bits/link $\times$ 7 switching paths) at FAST mode. The chip is implemented by 0.18-μm CMOS technology with 6-Al metal layers and its die area takes 5×5 mm^2. Fig. 19 shows the die photograph. The power dissipation of the on-chip network is 51 mW at FAST mode with full traffic condition. Fig. 20 shows the power reduction by each low-power technique and its breakdown. By using the proposed low-power techniques, the overall power consumption is reduced by 38% at maximum traffic condition.

The first silicon is successfully working and its on-chip packet signals are measured at FAST mode as shown in Fig. 21. The negative edge of the strobe signal is used for the timing reference of the $packet\langle 7{:}\,0\rangle$. A routing-switch recognizes the packet length instantly by sensing the end-of-packet (EOP) signal without parsing the packet-header which may consume additional packet delay. The waveforms of the SILENT coding are measured and shown in Fig. 22. In this example, the signal transitions on the channel are reduced almost by half due to the SILENT coding technique. Interestingly, the most significant bit, $Pkt\langle 7\rangle$, keeps silent and $Pkt\langle 6{:}\,4\rangle$ bits also become quiet after the coding. This is because image processing and 3-D Graphics applications have high data correlation. Graphics applications are demonstrated by the fabricated chip on the evaluation board, and it is shown in Fig. 23.

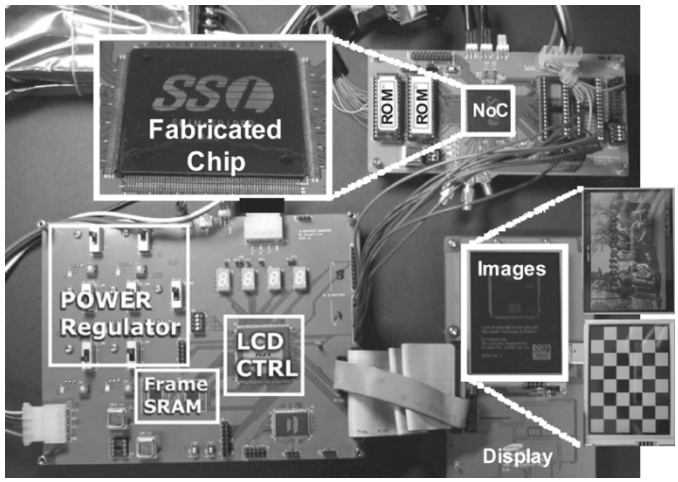


Fig. 23. Demonstration on the evaluation board.

V. Conclusion

A low-power NoC is designed and implemented for high-performance SoC applications. Heterogeneous IPs such as multiprocessors, memories, FPGA, and off-chip gateway with different timing references are interconnected in a hierarchical star topology. Various power-efficient techniques were suggested and implemented in each open system interconnection layer. Low-swing serial link and source-synchronous schemes in physical layer and low-energy serial link coding in data-link layer were proposed and realized on the NoC. Hierarchical circuit/packet switching, crossbar partial activation technique, and Mux-Tree based round-robin scheduler were also presented to reduce the power consumption in network layer. The on-chip network provides 11.2-GB/s bandwidth and consumes 51 mW at 1.6-GHz frequency. By using the proposed low-power techniques, the network power dissipation is reduced by 38%. The chip is implemented by 0.18-μm CMOS process and successfully operating with multimedia applications.

References

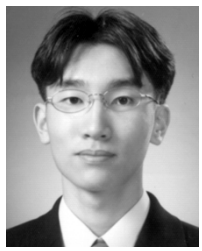
[1] International Technology Roadmap for Semiconductors [Online]. Available: http://public.itrs.net
[2] W. Dally *et al.*, "Route packets, not wires: On-chip interconnection networks," in *Proc. Des. Autom. Conf.*, Jun. 2001, pp. 684–689.
[3] L. Benini *et al.*, "Networks on chips: A new SoC paradigm," *IEEE Computer*, vol. 36, no. 1, pp. 70–78, Jan. 2002.

[4] D. Bertozzi *et al.*, "Xpipes: A network-on-chip architecture for gigascale system-on-chip," *IEEE Circuits Syst. Mag.*, vol. 4, no. 2, pp. 18–31, 2004.
[5] E. Rijpkema *et al.*, "Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip," in *Proc. Des., Autom. Test Europe Conf.*, Mar. 2003, pp. 350–355.
[6] V. Nollet *et al.*, "Operating-system controlled network on chip," in *Proc. Des. Autom. Conf.*, Jun. 2004, pp. 256–259.
[7] J.-S. Kim *et al.*, "On-chip network based embedded core testing," in *Proc. IEEE Int. SoC Conf.*, Sep. 2004, pp. 223–226.
[8] S.-J. Lee *et al.*, "An 800 MHz star-connected on-chip network for application to systems on a chip," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2003, pp. 468–469.
[9] M. Taylor *et al.*, "A 16-issue multiple-program-counter microprocessor with point-to-point scalar operand network," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2003, pp. 170–171.
[10] K. Lee *et al.*, "A 51 mW 1.6 GHz on-chip network for low-power heterogeneous SoC platform," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2004, pp. 152–153.
[11] H. Wang *et al.*, "A technology-aware and energy-oriented topology exploration for on-chip networks," in *Proc. Des., Autom. Test Europe Conf.*, Mar. 2005, pp. 1238–1243.
[12] BONE: Network-on-Chip Protocol [Online]. Available: http://ssl.kaist.ac.kr/ocn
[13] R. Ho *et al.*, "Efficient on-chip global interconnects," in *IEEE Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2003, pp. 271–274.
[14] C. Svensson, "Optimum voltage swing on on-chip and off-chip interconnect," *IEEE J. Solid-State Circuits*, vol. 36, no. 7, pp. 1108–1112, Jul. 2001.
[15] P. Gupta *et al.*, "Design and implementing a fast crossbar scheduler," *IEEE Micro*, vol. 19, no. 1, pp. 20–28, Jan./Feb. 1999.
[16] E. Shin *et al.*, "Round-robin arbiter design and generation," in *Proc. IEEE Int. Symp. Syst. Synthesis*, Oct. 2002, pp. 243–248.
[17] P. Landman *et al.*, "Architectural power analysis: The dual bit type method," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, no. 2, pp. 173–187, Jun. 1995.
[18] K. Lee *et al.*, "SILENT: Serialized low-energy transmission coding for on-chip interconnection networks," in *IEEE Int. Conf. Comput.-Aided Des. Dig. Tech. Papers*, Nov. 2004, pp. 448–451.
[19] R. Woo *et al.*, "A 210-mW graphics lsi implementing full 3-D pipeline with 264 Mtexels/s texturing for mobile multimedia applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 358–367, Feb. 2004.
[20] C. Kretzschmar *et al.*, "Why transition coding for power minimization of on-chip buses does not work," in *Proc. Des. Autom. Test Europe Conf. (DATE)*, Feb. 2004, pp. 512–517.
[21] M. R. Stan *et al.*, "Bus-invert coding for low-power I/O," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, no. 1, pp. 49–58, Mar. 1995.

Kangmin Lee (S'01) received the B.S. and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2000 and 2002, respectively. His M.S. degree work concerned the design and implementation of a 10 Gb/s port shared-bus packet switch with embedded DRAM. He is currently pursuing the Ph.D. degree in electrical engineering at KAIST.

His research concerns the theory, architecture, and VLSI design of high-speed network switches and on-chip interconnection networks. He has published more than 20 papers in international journals and conferences.

Mr. Lee received the Best Design Award and the Silver Prize at the 2002 and 2004 National Semiconductor IC Layout Design Contest, respectively. He also won the Outstanding Design Award at the 2005 IEEE Asian Solid-State Circuits Conference (A-SSCC) Design Contest. He has served as a member of the Technical Program Committees of Design, Automation and Test in Europe (DATE) Conference and Exhibition.



Se-Joong Lee (S'00–M'06) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1999, 2001, and 2005, respectively.

Since 1999, he has performed several industrial projects including development of network memory, high-speed switch, and 3-D Graphics processors. His major research topics include high-speed on-chip interconnect and networking in a system-on-chip. Currently, he is with Communication System Laboratories, Texas Instruments Incorporated, Dallas, TX.

Hoi-Jun Yoo (M'95–SM'04) graduated from the electronic department of Seoul National University, Seoul, Korea, in 1983 and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1985 and 1988, respectively. His Ph.D. degree work concerned the fabrication process for GaAs vertical optoelectronic integrated circuits.

From 1988 to 1990, he was with Bell Communications Research, Red Bank, NJ, where he invented the two-dimensional phase-locked VCSEL array, the front-surface-emitting laser, and the high-speed lateral HBT. In 1991, he became Manager of a DRAM design group at Hyundai Electronics and designed a family of fast-1 M DRAMs and synchronous DRAMs, including 256 M SDRAM. From 1995 to 1997, he was a faculty member with Kangwon National University. In 1998, he joined the faculty of the Department of Electrical Engineering at KAIST. In 2001, he founded a national research center, System Integration and IP Authoring Research Center (SIPAC), funded by the Korean government to promote worldwide IP authoring and its SOC application. From 2003 to 2005, he was the Project Manager for SoC in the Korea Ministry of Information and Communication. His current interests are SOC design, IP authoring, high-speed and low-power memory circuits and architectures, design of embedded memory logic, optoelectronic integrated circuits, and novel devices and circuits. He is the author of the books *DRAM Design* (Hongleung, 1996; in Korean) and *High Performance DRAM* (Sigma, 1999; in Korean).

Dr. Yoo received the Electronic Industrial Association of Korea Award for his contribution to DRAM technology in 1994, and the Korea Semiconductor Industry Association Award in 2002.