A 118.4 GB/s Multi-Casting Network-on-Chip With Hierarchical Star-Ring Combined Topology for Real-Time Object Recognition

Joo-Young Kim, *Student Member, IEEE*, Junyoung Park, *Student Member, IEEE*, Seungjin Lee, *Student Member, IEEE*, Minsu Kim, *Student Member, IEEE*, Jinwook Oh, *Student Member, IEEE*, and Hoi-Jun Yoo, *Fellow, IEEE*

***Abstract*—A 118.4 GB/s multi-casting network-on-chip (MC-NoC) is proposed as communication platform for a real-time object recognition processor. For application-specific NoC design, target traffic patterns are elaborately analyzed. Through topology exploration, we derive a hierarchical star and ring (HS-R) combined architecture for low latency and inter-processor communication. Multi-casting protocol and router are developed to accelerate one-to-many (1-to-N) data transactions. With these two main features, the proposed MC-NoC reduces data transaction time and energy consumption for the target object recognition traffic by 20% and 23%, respectively. The 350 k MC-NoC fabricated in a 0.13 μm CMOS process consumes 48 mW at 400 MHz, 1.2 V.**

***Index Terms*—Hierarchical star-ring combined topology, multi-casting, network-on-chip, object recognition.**

I. Introduction

As THE semiconductor technology scales down and the number of intellectual properties (IPs) on a single chip increases, communication among the IP blocks becomes a critical problem in system-on-chip (SoC) design. Conventional bus based communication architecture cannot continue to scale due to long signal propagation delay, lack of bandwidth, and large power consumption [1]. As an alternative, network-on-chip (NoC) that brings networking theories and methods to on-chip communication becomes a promising architecture with better performance and scalability than bus [1]–[4]. Recently, it is quickly replacing the conventional bus based communication platforms especially in multi-core based SoCs [5]–[7].

On the other hand, object recognition identifying database objects out of the input image is an essential technology for artificial vision applications such as face detection, auto-navigated vehicles, mobile intelligent robots, and surveillance systems [8], [9]. Since these applications have to seamlessly interact with output environments received from the video camera, real-time object recognition over 30 frame/sec (fps) is demanded. However, the object recognition process includes lots of computationally intensive image processing tasks such as image filtering, histogram, and database matching, so that most of recognition processors [10]–[14] adopt a multi-core approach to meet their real-time requirement. As communication architecture, they employ NoC in order to incorporate multiple IP blocks as well as satisfy their data bandwidth requirements. Among these NoCs, the two primitive ones [10], [11] just provide a simple connectivity to their several IPs without any consideration of specific data transactions. However, as the applied recognition algorithms get complicated and the number of IP blocks increases over ten [12]–[14], the processor's communication requirements becomes challengeable to satisfy. Whether the NoC solves communication requirements of the processor or not can affect the overall system performance. The memory centric NoC [15] of 81.6GOPS object recognition processor [12] supports data synchronization management and maximum search function among the IP blocks for real-time object recognition feature extraction based on scale invariant feature transform (SIFT) algorithm [9]. As a result, the processor achieves 16 fps object feature extraction for QVGA (320×240) video images. In a real-time object recognition processor [13], the dynamically reconfigurable NoC [16] is proposed for processor's SIMD/MIMD mixed mode operation. With a low latency crossbar switch containing image-express channel, the proposed NoC enables 22 fps object recognition for QVGA video images.

In this paper, we describe an application-specific NoC named multi-casting NoC (MC-NoC) to enable 60 fps object recognition at VGA (640×480) screen resolution [14]. Through elaborate traffic analysis and topology exploration, we derive its two main features: a hierarchical star and ring combined topology and broad/multi-casting capability. The hierarchical star and ring combined topology proposed for low latency data transfer and fast inter-processor data communication results in total 118.4 GB/s bandwidth with less than 3 switch hopping latency. For multi-casting implementation, a simple multi-casting protocol is proposed with the designated source and destination IPs while cost efficient network switch is developed for it. As a result, the proposed MC-NoC accelerates data communication by 20% as well as reduces energy consumption by 23% under the target object recognition traffic.

Manuscript received November 06, 2009; revised March 03, 2010; accepted March 09, 2010. Current version published June 25, 2010. This paper was approved by Guest Editor Stefan Rusu.

The authors are with the Division of Electrical Engineering, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: jooyoung.kim@kaist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2010.2048085

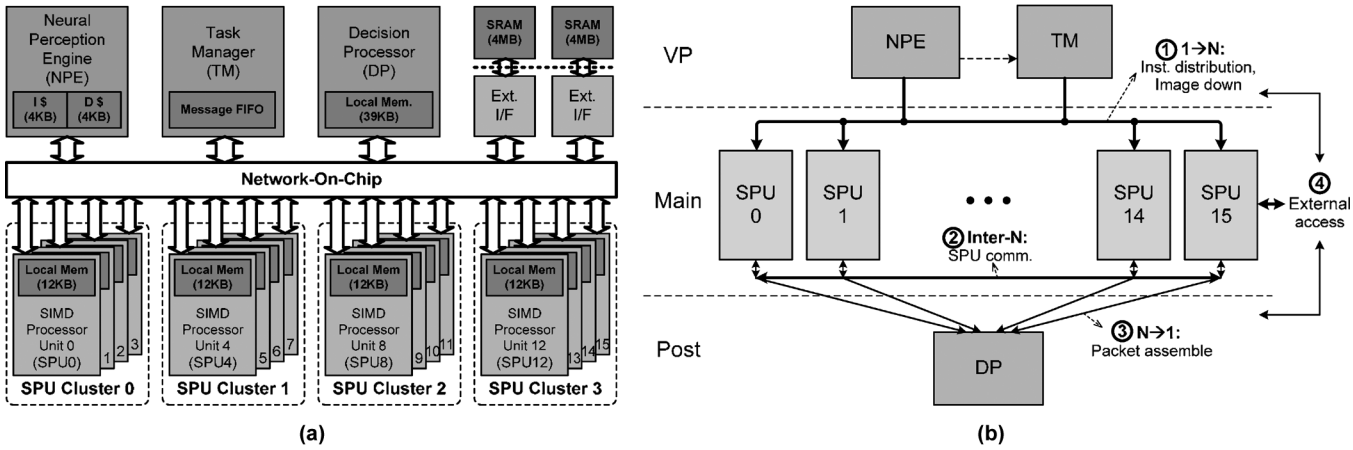


Fig. 1. (a) Proposed object recognition processor. (b) Major data transactions.

The rest of the paper is organized as follows. Section II analyzes traffic patterns of the target object recognition processor to derive its communication requirements. Based on these, Section III proposes the MC-NoC's hierarchical star and ring combined architecture via topology exploration. Section IV describes its wormhole routing based multi-casting protocol. Then, Section V describes detailed component design of the MC-NoC. After Section VI shows the chip implementation results, Section VI summarizes the paper.

II. Object Recognition Processor Architecture and Data Traffic Analysis

Fig. 1 shows the block diagram of the proposed object recognition processor [13] with its memory architecture and major data transactions. The proposed processor is developed with the target of 60 fps object recognition at VGA (640×480) video resolution for mobile robot applications. It consists of 21 IP blocks operating at 200 MHz frequency: a neural perception engine (NPE), a task manager (TM), 16 SIMD processor units (SPUs), a decision processor (DP), and two external interfaces. These 21 IPs are fully connected through a NoC whose operating frequency is twice the IP's, 400 MHz. Different from previous object recognition processors, the proposed processor divides the object recognition into three functional stages [17]: a visual perception (VP) computing regions-of-interest (ROIs) out of an input image, a main image processing (MIP) extracting local key-point vectors out of the tiled ROI area, and a post database matching (PDM) performing matching of the key-point vectors with a trained database for final decision. The NPE, 16 SPUs, DP are proposed for the three stages, respectively, while the TM is proposed to manage the ROI tasks of the 16 SPUs. In this object recognition processor, distributed memory architecture is adopted with two 200 MHz SRAM devices in off-chip domain. The NPE containing instruction and data cache initializes program codes of the other modules and then, each module utilizes its own local memory for intermediate data store when it performs the assigned tasks. Because every memory module is mapped to 32-bit address space in the top level, each IP can access any other memory modules in other IPs and two external ones through a fully interconnected NoC. With stream processing model [18], each IP minimizes external memory usage.

Fig. 1(b) shows major data transactions of the proposed processor. Three object recognition stages are mapped to a NPE, 16 SPUs, a DP, respectively, and are executed in the pipeline where the pipelined data are ROI tiles and key-point vectors between the first/second stage and second/third stage, respectively. The NPE sends the information of ROI tiles to the message FIFO in TM to make the TM assign them to 16 SPUs while the 16 SPUs store the resulting key-point vectors to the memory buffer in the DP. Due to the pipelined operation, lots of data transactions seamlessly occur among the IP blocks throughout the processing time. Under this condition, the NoC should provide both sufficient data bandwidth and low latency transfer to enable the processor's target 60 fps processing at VGA video resolution. Major data transactions are as follows. At the VP stage, the NPE distributes $>$0.8 Mb of instruction kernels to 16 SPUs. The TM, in between the NPE and 16 SPUs, assigns the delivered ROI tiles to the 16 SPUs and manage their task threads. In this process, the TM transfers $>$2 Mb of image pixel data to 16 SPUs each frame. In the MIP stage, the 16 SPUs perform the feature description algorithm to generate key-point vectors out of the selected ROI tiles. While running the algorithm, the multiple SPUs communicate $>$200 Kb intermediate data themselves and transfer $>$100 Kb key-point vectors to the DP each frame. In the PDM stage, the DP performs vector matching for the aggregated key-point vectors with a trained object database. From the three stages, each IP block requests $>$1 Mb raw image and intermediate data transactions with external memories each frame. Table I summarizes detailed traffic characteristics in the proposed processor.

Based on the obtained traffic patterns, we derived the following five NoC requirements for real-time object recognition. First of all, regardless of traffic patterns, low latency data transaction is required for real-time processing. Because every IP block of the proposed processor performs in-order processing, long data latency for read operations can seriously affect the overall performance. Second, 1-to-N data transactions are frequently used from the NPE/TM to 16 SPUs, e. g., when the NPE/TM distribute instruction kernels, ROI tile tasks, and image data to 16 SPUs. To accelerate these data

TABLE I
DETAILED TRAFFIC CHARACTERISTICS

Initiator	Destination	Required BW	Traffic characteristics		Description
NPE	EXT	Negligible	Uni-cast	No bursty	Boot-up process
NPE	16 SPUs	48Mb/s	Multi-cast	Bursty	Instruction distribution
NPE	TM	Negligible	Uni-cast	No bursty	ROI tile transfer
TM	16 SPUs	120Mb/s	Multi-cast	Bursty	image download
TM	16 SPUs	Negligible	Uni-cast	No bursty	SPU control
16 SPUs	16 SPUs	12Mb/s	Uni-cast	No bursty	Inter-SPU communication
16 SPUs	EXT	60Mb/s	Uni-cast	Bursty	Intermediate data load/store
16 SPUs	DP	6Mb/s	Uni-cast	Bursty	Key-point vector aggregation
NPE	EXT	Negligible	Uni-cast	No bursty	Recognition result out

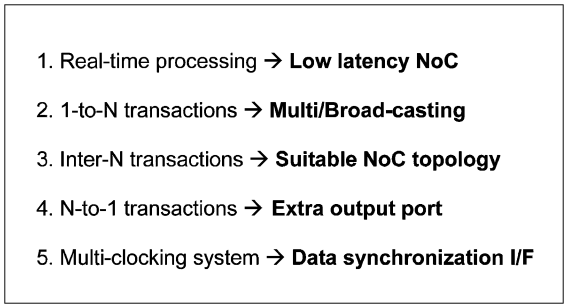


Fig. 2. Application specific NoC requirements.

transactions, broad- and multi-casting capability is demanded in the NoC. Third, sufficient communication channels are required among the SPUs for their co-processing. To this end, a suitable NoC topology should be investigated. Fourth, N-to-1 data transactions are required for the aggregation of vector packets from 16 SPUs to a DP. An additional output port is needed to DP to handle this problem. Lastly, the NoC should provide a multi-clocking environment for the overall SoC to allow each IP to use a preferable clock frequency. To this end, every interfacing region between the NoC and the IP blocks should have data synchronization function to traverse between heterogeneous clock domains. Fig. 2 summarizes the five application-specific NoC requirements of the proposed object recognition processor.

III. HIERARCHICAL STAR AND RING COMBINED TOPOLOGY

A. Topology Exploration

In the design of a NoC, topology selection deciding the structural organization of network switches is a primary concern. By the topology, not only the performance of NoC such as average latency and total bandwidth but also the implementation cost is roughly determined. To find out the most suitable NoC topology for the target object recognition, we choose four topologies depicted in Fig. 3: mesh [5], [6], hierarchical star [3], [4], [19], ring [7], and proposed hierarchical star-ring (HS-R) combined topology. The mesh topology has strength in regularity and scalability, but it is not good in terms of cost efficiency because each IP requires its own network switch. On the other hand, the hierarchical star topology is a cost efficient solution based on its tree structure that results in low latency data communication. However, the problem is that significant traffic workloads aggregate to network switches in higher level. The ring topology also has good cost efficiency, however, its long hopping latency cannot continue scaling to many-core era containing tens of IPs. The proposed HS-R topology combines the hierarchical star and ring topology for short latency data transmission and inter-processor data communication, respectively. Based on the hierarchical star networks, it adds ring networks to achieve shorter hopping latency as well as to mitigate workload aggregation to higher level network switches. Fig. 4 shows the block diagram of the conventional hierarchical star and proposed HS-R topology where the number of IPs and network hierarchies scale to 64 and 3, respectively. Different from the hierarchical star topology, the HS-R topology additionally interconnects the switches in the same network hierarchy with the ring topology. In this diagram, the number in each IP box and grouped IPs by the distance level means the switch hopping latency from the start IP $\boldsymbol{S}$ and the set of IPs having the same hopping latency from the start IP $\boldsymbol{S}$, respectively. The additional ring networks of the HS-R topology reduce the hopping latency from the start IP $\boldsymbol{S}$ by 1 when it accesses the IPs belongs to the neighbor clusters of each same distance group. As shown in Fig. 4, half of the overall IPs colored in dark gray are benefited by the HS-R topology.

To evaluate the performance of above four topologies under target application traffic, we made a cycle accurate NoC simulator broadly consisting of traffic model library and NoC model library, as shown in Fig. 5. The traffic model library contains synthetic traffic patterns and traffic generating subjects such as real IP's instruction set simulator and traffic generator. The traffic generator can generate some typical traffic patterns as well as arbitrary ones by reading a traffic description file (TDF) that compiles the time and data of each data transaction event. On the other hand, the NoC model library contains building components of NoC such as a network switch, communication socket, and network wrapper. We use transaction level modeling (TLM) [20] in the NoC model library. Because the TLM operates based on the existence of transactions, it is good to integrate various IP blocks modeled with different abstraction levels. In addition, its event driven characteristic is beneficial

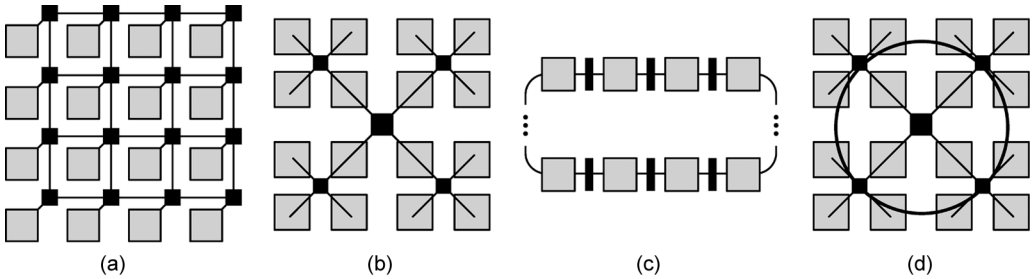


Fig. 3. NoC topologies. (a) Mesh; (b) H-star; (c) ring; (d) H-star + ring.

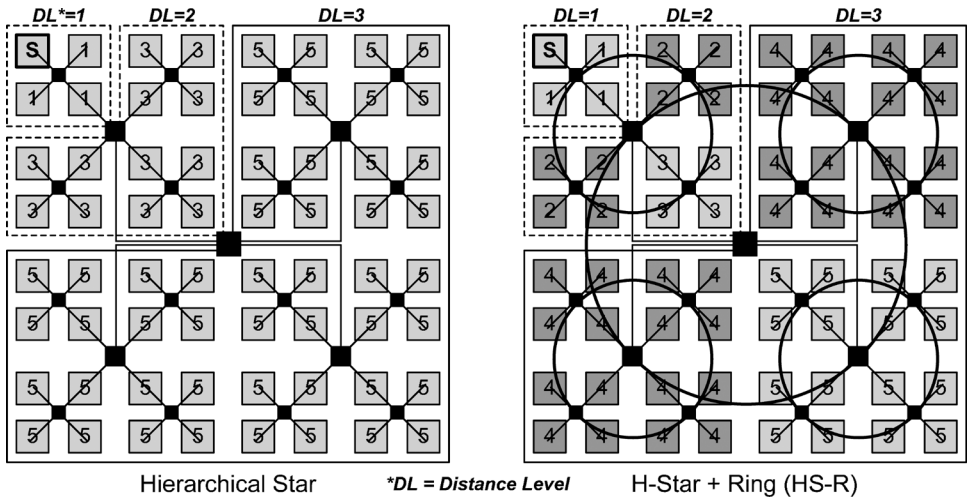


Fig. 4. Comparison between hierarchical star and proposed hierarchical star–ring combined topology.

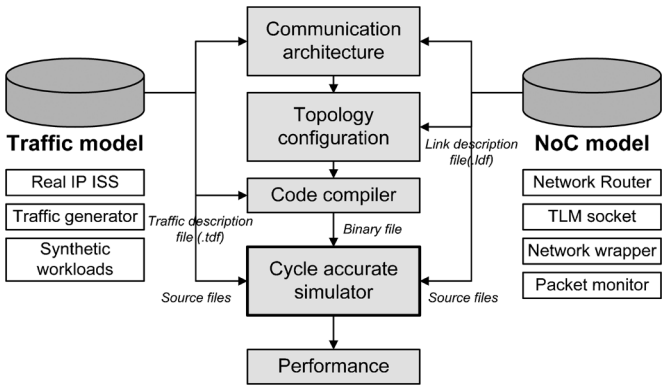


Fig. 5. NoC simulation framework.

for fast simulation time. A monitoring unit is included in each network packet and switch model for packet tracing and statistic extractions. Using these two model libraries, we can constitute a cycle accurate NoC simulation framework. The traffic generating IP blocks are connected to the NoC using network wrapper and TLM socket interface, while the NoC topology is built by a link description file (LDF) that describes the connection status of whole network switches.

Using the implemented NoC framework, we measured the average latency, bandwidth utilization, and implementation cost of the four NoC topologies when the target three-stage object recognition algorithm runs for several VGA (640 $\times$ 480 pixels) input images. Average 56.49 s is spent by the simulator to process a single VGA image. In this experiment, cycle accurate models are used for the TM, 16 SPUs, and DP while a functional model is used for the NPE. The bandwidth utilization is calculated by dividing the overall amount of packet data through every network switch by the maximum total bandwidth of every switch. The cost of the NoC topology is measured by the amount of employed network switches and interconnecting wires. Assuming every network switch is square (same input and output port number, e. g., 5 $\times$ 5, 6 $\times$ 6, and 7 $\times$ 7), the implementation costs of a network switch itself and the wires from it are proportional to the number of input ports. Therefore, the implementation cost of the overall NoC can be represented by following equation where α, β, γ means the cost of a 1 $\times$ 1 sized switch, the cost of a single wire in average length, and the data bit-width of each port.

$$\text{Cost} = \left(\sum_{All\ switches} \#\ \text{of input ports} \right) \times (\alpha + \beta \cdot \gamma)$$

Fig. 6 shows the experimental results. The implementation cost is represented by the total number of input ports from all switches, a normalized value with the unit cost of a switch and wires per port. For latency, the hierarchical star and HS-R topology show better performances than the mesh and ring topology because their tree-like structure logarithmically reduces the switch hopping number to access other IP blocks. The HS-R topology further reduces the latency from the hierarchical star topology by accelerating inter-processor communication via supplemented ring networks. For bandwidth utilization,

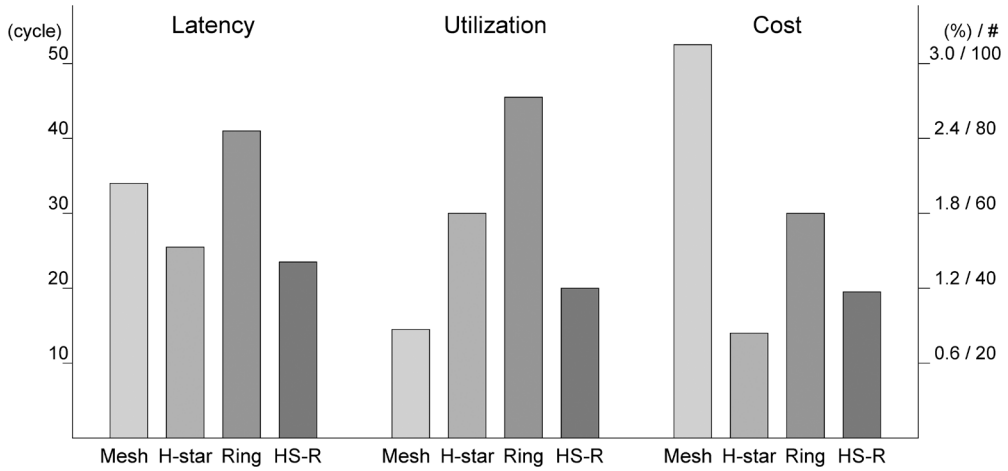


Fig. 6. Topology exploration results.

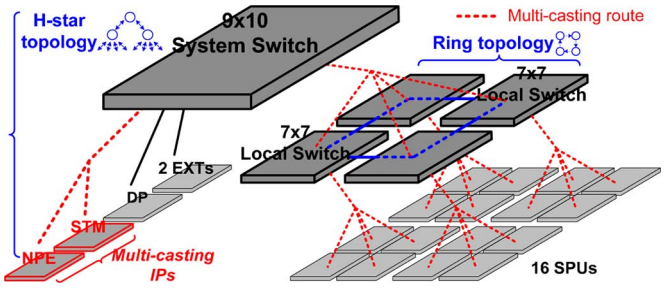


Fig. 7. MC-NoC architecture.

the mesh topology is the lowest among the four topologies because it has the largest maximum total bandwidth due to its abundant network switches and links. However they also bring the most expensive implementation cost to mesh topology. For implementation cost, the hierarchical star and HS-R show good efficiency. In conclusion, mesh is not a suitable topology for the target traffic because it does not show better latency performance than the others even it requires larger implementation cost. The ring topology seems to be the worst among the four topologies. This is because its expected switch hopping number becomes too large in these 21 IP SoC. For smaller sized SoCs, the ring topology can be a good solution with its low cost interconnection. We can decide that the proposed HS-R topology is the most suitable topology for the target traffic with the lowest data latency and supplemented data bandwidth for inter-processor communication.

B. MC-NoC Architecture

Fig. 7 shows the overall architecture of the proposed MC-NoC consisting of a 9×10 system network switch and four 7×7 local network switches. The 16 SPUs are connected to the system network through the four local network switches while the rest of IP blocks such as the NPE, TM, DP, and two external interfaces are directly connected to the system network switch. As decided in the previous section, the MC-NoC exploits the HS-R combined topology. It adopts a hierarchical star topology as a basic starting topology for low latency data communication, and then, supplements a ring topology to the local networks for high speed inter-SPU data transactions. The additional ring networks of the combined topology provide a maximum bandwidth of 25.6 GB/s (4 links $\times$ (2 input ports $+$ 2 output ports)$\times 4$ byte $\times$ 400 MHz, assuming all physical channels transfer data) between the local networks, and allow each SPU to access the other SPUs in neighbor local networks in two switch hops without accessing the system network. In addition, for N-to-1 data transactions, the output port of the DP is modified to a dual port. One of them is dedicated to key-point vector packet aggregation from the 16 SPUs. In overall, the topology combined MC-NoC provides overall 118.4 GB/s ($(9$ input ports$+9$ output ports$)+4\times(7$ input ports$+$ 7 output ports$) \times 4$ byte $\times$ 400 MHz) total bandwidth with the switch hopping latency of less than 3.

IV. Multi-Cast Protocol

Multi-casting transmitting the same packet to multiple destinations [21] is an effect method when a single source wants to send the identical packets to multiple destinations. It is also very useful for the 1-to-N data requirements of the proposed processor such as program code distribution and image data download. However, multi-casting operation in NoC can cost a lot because protocol, routing, and network switch design should be redesigned. In this chapter, we present a cost efficient multi-casting protocol and implementation in the proposed MC-NoC based on the fact that the multi-casting source and destination IP are already determined.

Fig. 8 shows the block diagram of the network switch consisting of input ports containing queuing buffer, output arbiters, crossbar switch, and output ports. It uses a wormhole routing protocol that controls the packet operation in a smaller flow control unit (flits) to reduce the size of input buffer. To achieve high throughput, the network switch performs a packet transmission with four pipeline stages. First, the incoming packet is buffered at the input port in a flit unit. Then, each input port sends request and obtain grant to/from the output arbiter. After getting a grant, the packet traverses the crossbar switch. To prevent the packet loss, the network switch uses a credit based back pressure scheme [22] that controls the flits in the router by the congested

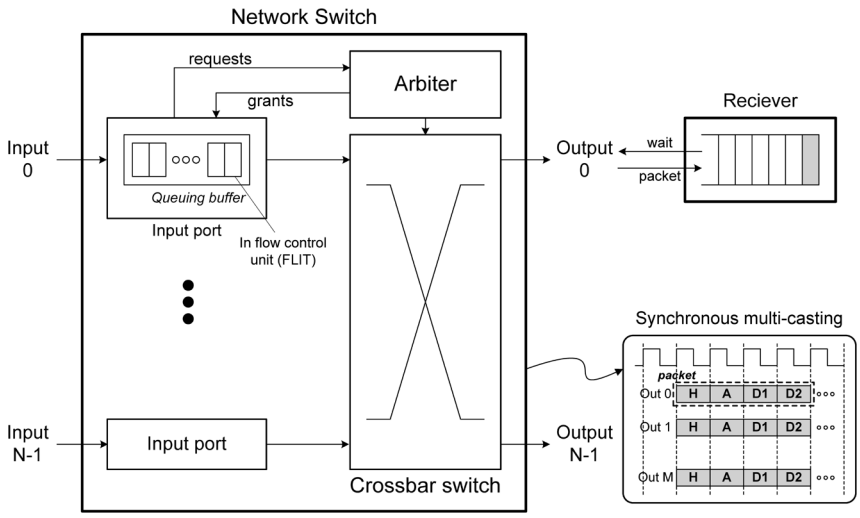


Fig. 8. Network switch block diagram.

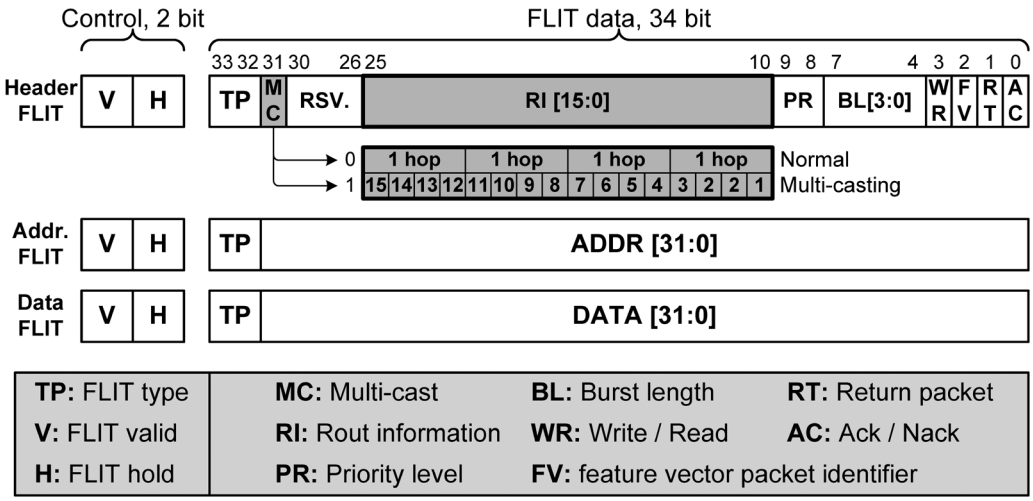


Fig. 9. Packet format of MC-NoC.

status of receiver side. For multi-casting operation, it uses a synchronous method [23] that transmits the packet to the multiple output ports at the same time. By eliminating packet retransmission, it can achieve simpler routing algorithm as well as higher energy efficiency than asynchronous method.

Fig. 9 shows the detailed packet format of the MC-NoC. It consists of three flit types—header, address, and data—while each contains 34-bit information including 2-bit type identifier. The flit is efficiently controlled by 2-bit flow control signals, valid and back pressure hold signal.

The header flit contains all information of the packet such as read/write, 4-bit burst length, 2-bit priority level, and 16-bit routing information (RI). The address flit indicates the address of the final destination IP where the data flits are read or written. When the packet is routed in each network switch, the header filt forwards first and the address/data flits follow after it. The burst length information enables burst data transactions up to 8 data flits. For this, the write burst packet sends header, address, and multiple data flits of burst length to the target IP while the read burst packet sends only header and address flit to get multiple data flits of burst length from the target IP. The 2-bit priority level indicates the priority of packet in the situation that the network has contentions. By obtaining the arbiter's grant earlier in each network switch, the packet with the higher priority level is more quickly routed than the packet with the normal priority level. Lastly, the 16-bit source defined routing information (RI) describes where the packet should be routed in each network switch to the final destination. It allows 4 switch traversals for normal packets and a multi-casting to arbitrary SPUs for multi-casting packets. For multi-casting case, we use a bit string encoding [24] to notify the destination SPUs. In this encoding, each bit of 16-bit RI corresponds to each SPU and configuring a bit to 1 means the SPU is one of the multi-casting destinations. Without describing detailed routing path, the 16-bit RI efficiently implicates the multi-casting destinations out of 16 SPUs. Each network switch decodes the 16-bit multi-casting RI to generate correct routing paths to multiple destinations.

Fig. 10 shows the decoding process of a multi-casting packet in the MC-NoC. With a bit string encoding, the example RI intends to send a packet to SPU 0, 2, 3, 13, and 15. To this end, the MC-NoC adopts a hierarchical approach using its hierarchical-star topology: the system network performs the first multi-casting to four SPU local networks, and then, local network perform the second multi-casting to final destination

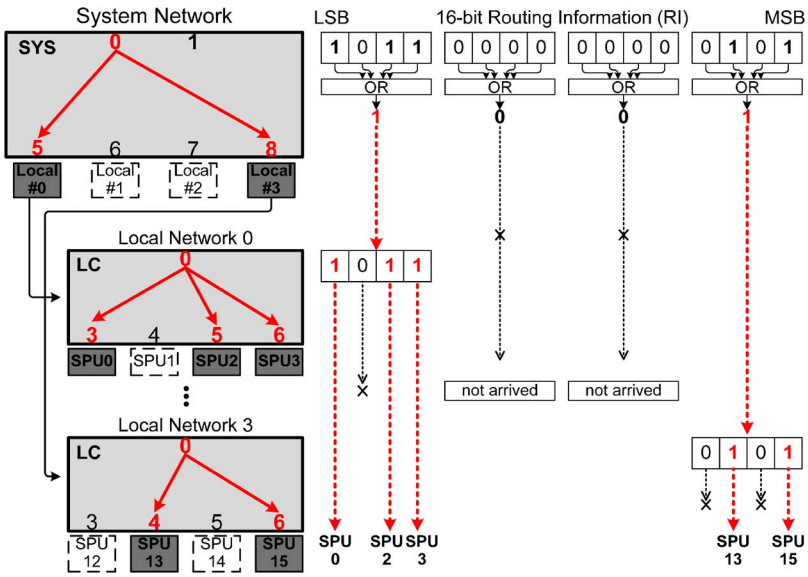


Fig. 10. Multi-casting process of MC-NoC.

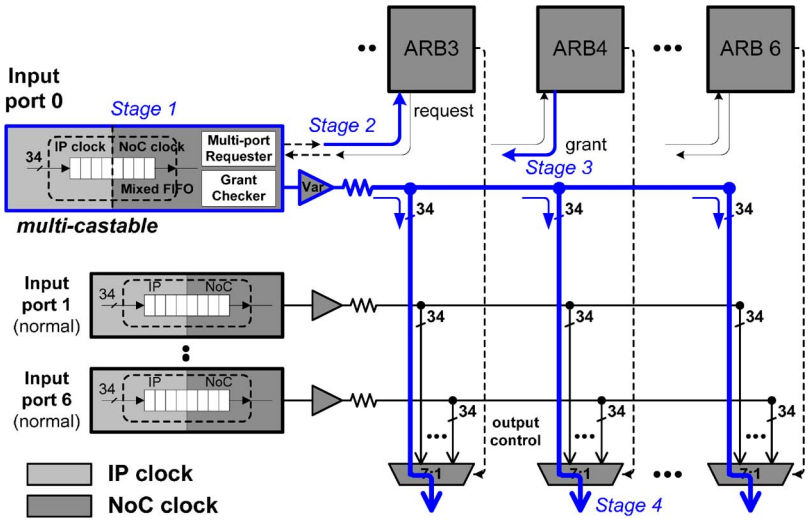


Fig. 11. Four-stage pipelined multi-casting network switch.

SPUs. In the first multi-casting, the system network sends the packet to the local networks containing at least one destination SPU. The destination local networks can be identified by performing OR operation on each of four 4-bit RI segments. In the second multi-casting, each destination local network sends a packet to destination SPUs according to corresponding 4-bit RI segment. Through this two step multi-casting, the packet correctly arrives at the destination SPUs. Different from the mesh based NoC should consider the optimized multi-cast routes out of lots of possible candidates, the hierarchical star based MC-NoC can perform the hierarchical and deterministic multi-cast routing because its upper level network switches interconnect all of the lower level network switches. This hierarchical multi-casting method can be expanded to larger MC-NoCs having hierarchies more than two.

V. Detailed MC-NoC Design

In this Section, we discuss real implementation of MC-NoC. The MC-NoC has three design issues to address. First, each network switch should be able to perform multi-casting for overall multi-casting capability. Second, the interface between the MC-NoC and IP blocks can convert the heterogeneous clock domains for multi-clocking SoC platform. Third, the MC-NoC should operate with low power consumption for low overhead interconnection platform.

A. Multi-Casting Network Switch

Fig. 11 depicts the block diagram of multi-casting network switch composed of input ports, output arbiters, crossbar fabric, and output ports. It performs packet transmission with four pipelined stages. First, the incoming flits are buffered at 8-depth 32-bit first-in-first-out (FIFO) queue containing synchronization interface for the two heterogeneous clock domains. Then, each activated input port sends a request signal to the output arbiter to get a grant signal to traverse the crossbar fabric. The arbiters perform round-robin scheduling with two priority levels. After getting a grant signal, the input port sends the packet to the destination output port through a fully connected crossbar fabric. In case of multi-casting, the input port sends multiple request signals to multiple output arbiters to ensure

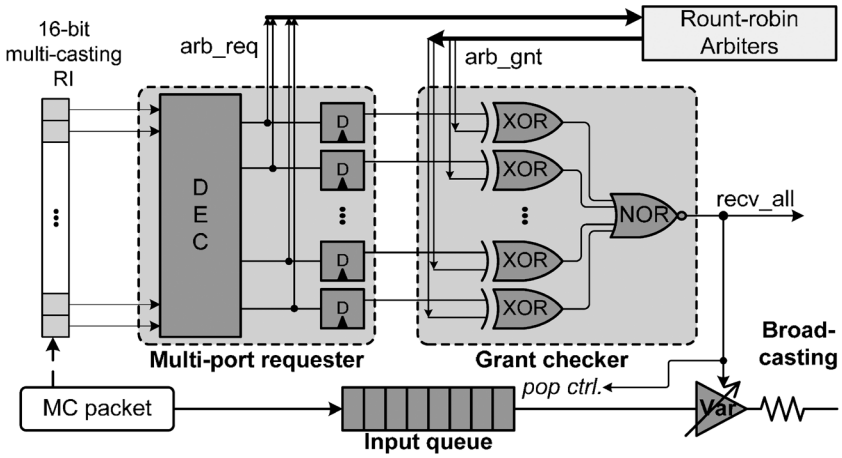
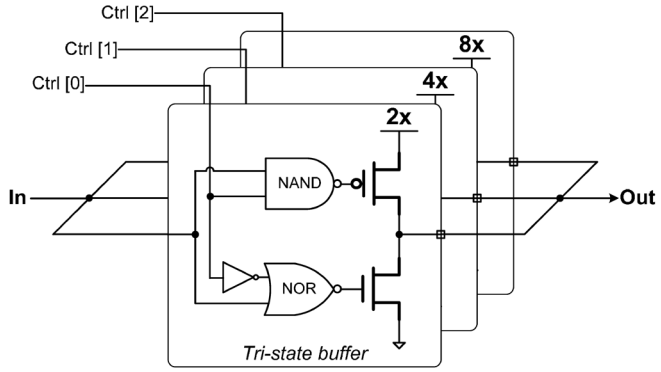


Fig. 12. Multi-casting port.



Ctrl[2:0]	# of Dest. Outputs	Driver Strength	Power Consumption
001	1	2x	0.108mW
010	2	4x	0.169mW
011	3	6x	0.298mW
100	4	8x	0.338mW
101	5	10x	0.472mW
110	6	12x	0.528mW
111	7, 8	14x	0.659mW / 0.712mW

Fig. 13. Circuit diagram of variable strength driver.

all grant signals of desired output ports. The multi-casting input port waits until all grant signals are returned. For this operation, the multi-casting input port employs a multi-port requester and grant checker shown in Fig. 12. The multi-port requester decodes the 16-bit routing information, generates request signals to multiple arbiters, and latches the request signals in a 16-bit register. Meanwhile, the grant checker holds the multi-casting packet in the FIFO queue until the received grant signals become equal to the latched request ones. After gathering all grant signals, the multi-casting is performed using broad-casted wires of crossbar fabric. In case of arbitrary destinations are blocked by some reasons, the multi-casting port should select whether it hold the grants until to gather all grants or withdraw the request and request later. In this design, the multi-casting request is withdrawn when the other inputs send requests to the output arbiters whose grants are already captured by the multi-casting port which is waiting remaining grants, not performing multi-casting. In the proposed processor because the multi-casting is used in cases that the NPE performs multi/broad-casting program code to 16 SPUs at the beginning of the processor operation and the TM performs multi-casting the image data to two SPUs, above conflicts in multi-casting rarely happen and barely affect the overall performance.

To provide sufficient driving strength for variable loads in multi-casting, we specially devise a variable strength driver whose circuit diagram is shown in Fig. 13. It consists of three parallel tri-state buffers whose driving forces are different to $2\times$, $4\times$, and $8\times$, respectively, while their output wires are connected into a single final output to enhance the overall circuits output driving strength. Using 3-bit control signals that each of them decides the activation of each tri-state buffer, the variable strength driver can dynamically configure its output driving strength from $2\times$ to $14\times$ by a $2\times$ step according to the number of output ports of multi-casting. The table in Fig. 13 shows the operation modes of the variable strength driver by the control signals and its energy consumptions according to various multi-casting output loads.

B. FIFO Based Synchronization Interface

To allow the heterogeneous clock frequencies between the IPs and MC-NoC, the first-in-first-out (FIFO) based synchronization is performed in every input/output port of network switches interfacing with IP blocks. Fig. 14 shows the block diagram of FIFO based synchronization interface that exists in each input port. It consists of an 8-depth FIFO buffer, write control block, and read control block, which are for data storage, write and read operation control and pointer management, respectively. A flag detector, in between the write and read control block, generates combinational flag signals such as empty and full using relative locations of the two pointers. When the detected empty or full signal is used in the other clock domain for flow control signal generation like request or hold, they are synchronized with the clock of the other domain using a pipelined synchronization method [25].

C. Fine-Grained Clock Gating

For low power consumption, the MC-NoC performs a fine-grained clock gating that activates the only required packet routing path in each network switch. Fig. 15 shows the block diagram of fine-grained clock gating and its operating timing diagram. The monitor unit in each network switch detects the

TABLE II
CHIP SUMMARY

Process Technology	0.13mm 1P 8M CMOS
Die Size	7mm x 7mm
Power Supply	1.2V core, 2.5V I/O
Operating Frequency	200MHz IPs / 400MHz NoC
Transistor Counts	36.4M transistors 3.73M gates / 396KB SRAM
Power Consumption	496mW (average)
Topology	Hierarchical star + Ring
Transistor Counts	350k gates
Total Bandwidth	118.4 GB/s (H-star:92.8GB/s, Ring:25.6GB/s)
Latency	Less than 3 switch hops
Operating Frequency	400MHz
Power dissipation	48mW
Protocol	Wormhole routing / Multi-casting protocol Burst packet transmission (up to 8 flits) 2 priority levels
Queuing model	Input queuing (depth 8)
Interface	Heterogeneous clock interface (FIFO based synchronization)

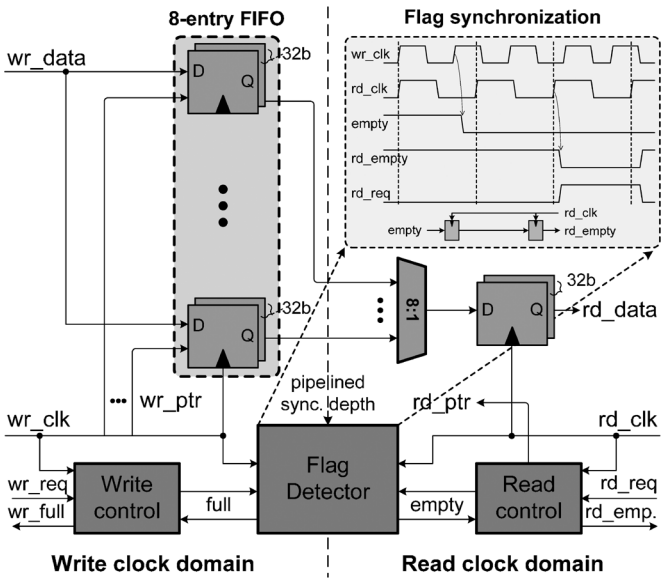


Fig. 14. Heterogeneous clock interface.

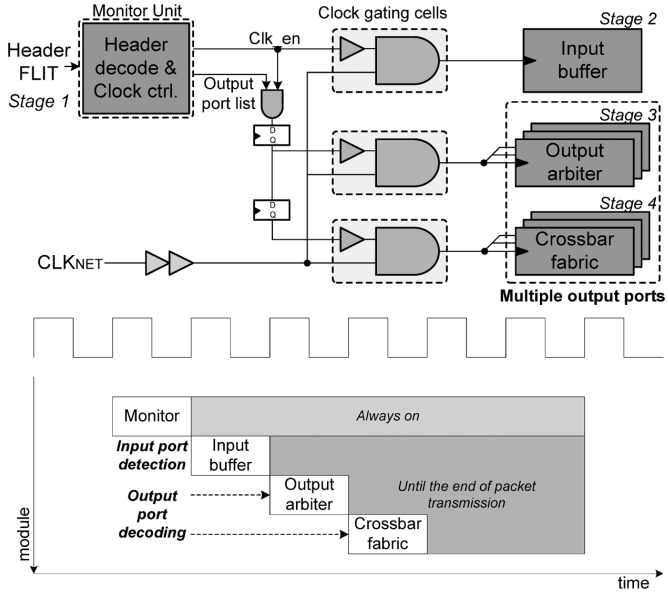


Fig. 15. Fine grained clock gating.

existence of incoming flits to activate corresponding input queuing buffers. In addition, it partially decodes the header flit to generate the list of output ports to activate the output arbiters and crossbar fabric. Because the proposed MC-NoC supports multi-casting operation, the destination output ports can be multiple. With a simple D-flip-flop chain, the activation time of each switch component such as input queues, output arbiters, and crossbar fabric is determined in a pipelined manner. The activated components operate until the end of a single packet transmission while the monitor unit is always turned on. By controlling the activation of each switch component in cycle level including the input buffers consuming almost 90% of network switch power, the fine-grained clock gating can save 28% power on average.

VI. IMPLEMENTATION RESULTS

To evaluate the performance of the MC-NoC, we measured the cycle count and energy consumption from gate-level simulation for the all packets transmitted during the target object recognition algorithm is running for a single VGA (640 × 480) video image. We also performed the identical experiments on a basic hierarchical star NoC without any scheme and a HS-R combined NoC without multi-casting capability for comparison. Fig. 16 shows the cycle count and energy reduction effect by the two main features of the MC-NoC. The HS-R combined topology contributes to inter-SPU data communication. By using the supplemented ring networks in inter-SPU communication, it shortens the routing path and reduces the number

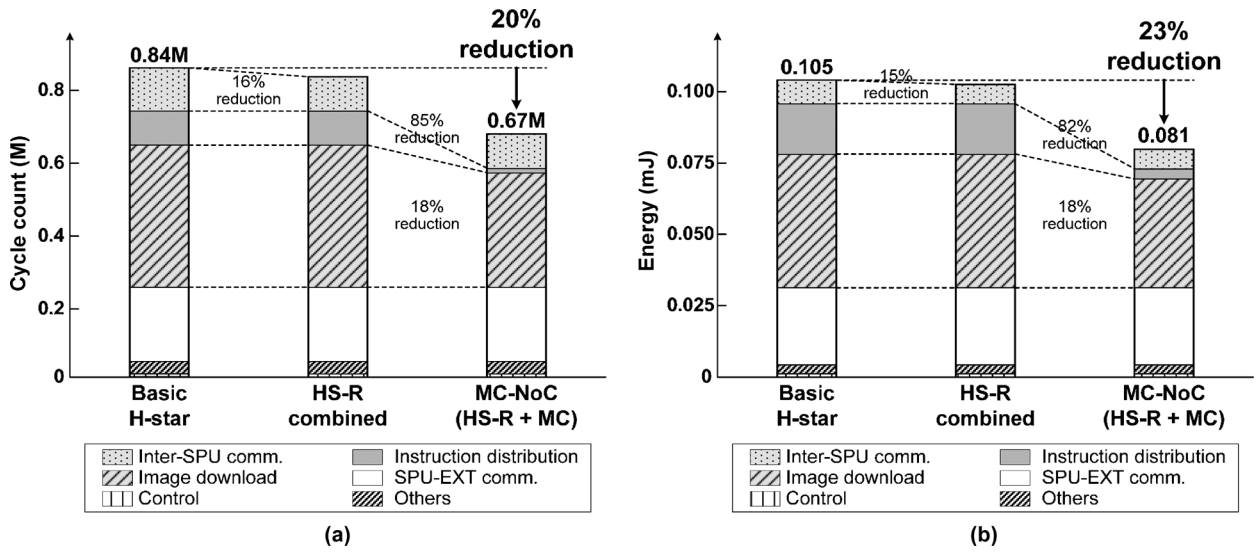


Fig. 16. MC-NoC evaluation results. (a) Cycle count. (b) Energy consumption.

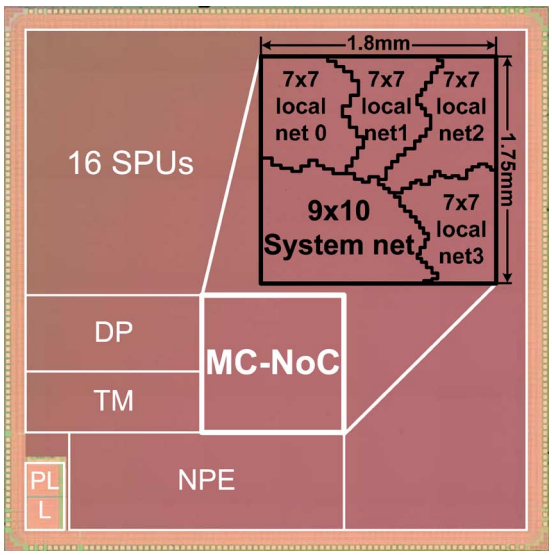


Fig. 17. Chip micrograph.

of packet buffering. The multi-casting capability contributes to program distribution and image download. By removing retransmissions of the same packet to multiple IPs, it not only significantly reduces the number of packet buffering but also accelerates the data transactions as well. As a result, the MC-NoC reduces the overall cycle count and energy consumption in data communication by 20% and 23%, respectively, compared with a basic hierarchical-star NoC.

The real-time object recognition processor with the proposed MC-NoC is fabricated in a 0.13 μm eight-metal CMOS process. Fig. 17 shows the chip micrograph. The overall recognition processor has 49 mm^2 die area with 3.73 M gates while the MC-NoC accounts for 350 k gates. The operating frequency is 200 MHz for IP blocks and 400 MHz for MC-NoC. The MC-NoC provides 118.4 GB/s total bandwidth (92.8 GB/s from hierarchical star networks, 25.6 GB/s from ring networks) at 400 MHz frequency and supports multi-casting to 16 SPUs. The overall processor consumes 496 mW at 1.2 V while the object recognition is running at 60 frame/sec for VGA video input. Under this condition, the MC-NoC dissipates 48 mW, only 9.7% of the overall processor. Table II summarizes the chip and MC-NoC features.

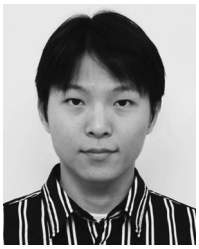
VII. Conclusions

A 118.4 GB/s multi-casting network-on-chip (MC-NoC) is developed as communication platform for a real-time object recognition processor. Through topology exploration, we derive the HS-R combined topology for low latency and inter-processor data communications. Multi-casting is supported to accelerate 1-to-N data transactions. With these two main features, the proposed MC-NoC reduces data transaction time and energy consumption by 20% and 23%, respectively, under the target object recognition traffic. The MC-NoC is fabricated in a 0.13 μm CMOS process in the proposed object recognition processor while consuming 48 mW at 400 MHz, 1.2 V.

References

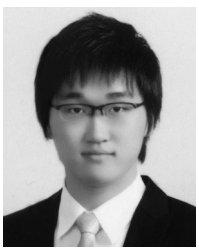
[1] L. Benini and G. D. Micheli, "Networks on chips: A new SoC paradigm," *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.
[2] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. 38th Design Automation Conf.*, Jun. 2001, pp. 684–689.
[3] S.-J. Lee *et al.*, "An 800 MHz star-connected on-chip network for application to systems on a chip," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2003, pp. 468–469.
[4] 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. (ISSCC) Dig. Tech. Papers*, Feb. 2004, pp. 152–153.
[5] S. Vangal *et al.*, "An 80-tile 1.28TFLOPS network-on-chip in 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2007, pp. 98–99.
[6] S. Bell *et al.*, "TILE64TM processor: A 64-core SoC with mesh interconnect," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2008, pp. 88–89.
[7] T. W. Ainsworth and T. M. Pinkston, "Characterizing the cell EIB on-chip network," *IEEE Micro*, vol. 27, no. 5, pp. 6–14, 2007.
[8] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. New York: Prentice Hall, 2002.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *ACM Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Jan. 2004.
[10] R. R. Rojas *et al.*, "Object recognition SoC using the support vector machines," *J. Appl. Signal Process.*, pp. 993–1004, 2005.
[11] H.-C. Lai *et al.*, "Communication-aware face detection using NoC architecture," in *Proc. Int. Conf. Computer Vision Systems (ICVS)*, 2008, pp. 181–189.
[12] D. Kim *et al.*, "An 81.6 GOPS object recognition processor based on NoC and visual image processing memory," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2007, pp. 443–446.
[13] K. Kim *et al.*, "A 125GOPS 583 mW network-on-chip based parallel processor with bio-inspired visual attention engine," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 136–147, Jan. 2009.
[14] J.-Y. Kim *et al.*, "A 201.4GOPS 496 mW real-time multi-object recognition processor with bio-inspired neural perception engine," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2009, pp. 150–151.
[15] D. Kim *et al.*, "Implementation of memory-centric NoC for 81.6 GOPS object recognition processor," in *Proc. IEEE A-SSCC*, Nov. 2007, pp. 47–50.
[16] K. Kim *et al.*, "A 76.8 GB/s 46 mW low-latency network-on-chip for real-time object recognition processor," in *Proc. IEEE A-SSCC*, Nov. 2008, pp. 189–192.
[17] J.-Y. Kim *et al.*, "Real-time object recognition with neuro-fuzzy controlled workload-aware task pipelining," *IEEE Micro*, vol. 29, no. 6, pp. 28–43, Nov./Dec. 2009.
[18] W. J. Dally *et al.*, "Stream processors: Programmability with efficiency," *ACM Queue*, vol. 2, no. 1, pp. 52–62, 2004.
[19] K. Lee *et al.*, "Low-power networks-on-chip for high-performance SoC design," *IEEE Trans. VLSI*, vol. 14, no. 2, pp. 148–160, Feb. 2006.
[20] F. Ghenassia, *Transaction Level Modeling With SystemC: TLM Concepts and Applications for Embedded Systems*. New York: Springer, 2005.
[21] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, ser. The Morgan Kaufmann Series in Networking. : Morgan Kaufmann, 2007.
[22] H.-J. Yoo, K. Lee, and J. K. Kim, *Low Power NoC for High Performance SoC Design*. Boca Raton: CRC Press, 2008.
[23] C. M. Chiang and L. M. Ni, "Deadlock-free multi-head wormhole routing," in *Proc. First High Performance Computing–Asia*, Taiwan, Taiwan, Sep. 1995.
[24] C. Chiang and L. Ni, "Multi-address encoding for multicast," in *Proc. 1st Int. Workshop on Parallel Computer Routing and Communication, PCRCW'94*, Seattle, WA, May 1994, pp. 146–160.
[25] J. N. Seizovic, "Pipeline synchronization," in *Proc. IEEE ASYNC*, Nov. 1994, pp. 87–96.



Joo-Young Kim (S'05) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2005, 2007, and 2010, respectively.

Since 2006, he has been involved with the development of the parallel processors for computer vision. Currently, he is a postdoctoral researcher at KAIST. His research interests include multi-core architecture, network-on-chip, and VLSI implementations for computer vision applications.



Junyoung Park (S'09) received the B.S. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2009 and is currently working toward the M.S. degree in electrical engineering and computer science at KAIST.

Since 2009, he has been involved with the development of the parallel processors for computer vision. Currently, his research interests are many-core architecture and VLSI implementation for bio-inspired vision processor.

Seungjin Lee (S'06) received the B.S. and M.S. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2006 and 2008, respectively. He is currently working toward the Ph.D. degree in electrical engineering and computer science from KAIST.

His previous research interests include low power digital signal processors for digital hearing aids and body area communication. Currently, he is investigating parallel architectures for computer vision processing.

Minsu Kim (S'07) received the B.S. and M.S. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2007 and 2009, respectively. He is currently working toward the Ph.D. degree in electrical engineering and computer science at KAIST.

His research interests include network-on-chip based SoC design and bio-inspired VLSI architecture for intelligent vision processing.

Jinwook Oh (S'08) received the B.S. degree in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 2008. He is currently working toward the M.S. degree in electrical engineering and computer science at KAIST, Daejeon, Korea.

His research interests include low-power digital signal processors for computer vision. Recently, he has been involved with the VLSI implementation of neural networks and fuzzy logics.

Hoi-Jun Yoo (M'95–SM'04–F'08) graduated from the Electronic Department of Seoul National University, Seoul, Korea, in 1983 and received the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 1985 and 1988, respectively. His Ph.D. 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 from fast −1 M DRAMs and 256 M synchronous DRAMs. In 1998, he joined the faculty of the Department of Electrical Engineering at KAIST and now is a full Professor. From 2001 to 2005, he was the Director of System Integration and IP Authoring Research Center (SIPAC), funded by Korean government to promote worldwide IP authoring and its SOC application. From 2003 to 2005, he was the full time Advisor to Minister of Korea Ministry of Information and Communication and National Project Manager for SoC and Computer. In 2007, he founded SDIA(System Design Innovation and Application Research Center) at KAIST to research and develop SoCs for intelligent robots, wearable computers and bio systems. His current interests are high-speed and low-power network-on-chips, 3-D graphics, body area networks, biomedical devices and circuits, and memory circuits and systems. He is the author of the books *DRAM Design* (Seoul, Korea: Hongleung, 1996; in Korean), *High Performance DRAM* (Seoul, Korea: Sigma, 1999; in Korean), and chapters of *Networks on Chips* (New York, Morgan Kaufmann, 2006).

Dr. Yoo received the Electronic Industrial Association of Korea Award for his contribution to DRAM technology the 1994, Hynix Development Award in 1995, the Korea Semiconductor Industry Association Award in 2002, Best Research of KAIST Award in 2007, Design Award of 2001 ASP-DAC, and Outstanding Design Awards of 2005, 2006, 2007 A-SSCC. He is a member of the executive committee of ISSCC, Symposium on VLSI Circuits, and A-SSCC. He is the TPC chair of the A-SSCC 2008.