Adding Slow-Silent Virtual Channels for Low-Power On-Chip Networks

Hiroki Matsutani[1], Michihiro Koibuchi[2], Daihan Wang[1], and Hideharu Amano[1]

[1]Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama,
JAPAN 223-8522
{matutani,wang,hunga}@am.ics.keio.ac.jp

[2]National Institute of Informatics
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo,
JAPAN 101-8430
koibuchi@nii.ac.jp

Abstract

In this paper, we introduce the use of slow-silent virtual channels to reduce the switching power of on-chip networks while keeping the leakage power small. Adding virtual channels to a network improves the throughput until each link bandwidth is saturated. This enables us to reduce the switching power of on-chip networks by decreasing their operating frequency and supply voltage. However, adding virtual channels increases the leakage power of routers as well as the area due to their large buffers; so the runtime power gating is applied to individual virtual channels to eliminate this problem. We evaluate the performance of slow-silent virtual channels by using real application traces, and their power consumption (switching and leakage) is evaluated based on the detailed design of a virtual-channel router placed and routed with a 90nm technology. These evaluation results show that a network with three or four virtual channels achieves the best energy efficiency in a uniform traffic. In the cases of neighboring communications, a network with two virtual channels is better than the other networks with more virtual channels, because the performance improvement from no virtual channel to two virtual channels is the largest and their frequency and supply voltage can also be reduced well in these cases.

1 Introduction

Many studies have been conducted on Network-on-Chips (NoCs) [3][5][23] to connect a number of processing cores on a single chip by introducing a packet-switched network structure. NoCs have been utilized not only in high-performance microarchitectures, but also in cost-effective embedded devices mostly used in consumer equipments. These embedded applications usually require low power, since power consumption is the dominant factor on their battery life, heat dissipation, and packaging cost.

The overall power consumption consists of dynamic switching power and static leakage power. Switching power is still the major component of the overall power consumption during active operations; thus it should be reduced first. In addition, we need to take care of the leakage power, since it has already been consuming a substantial portion of the active power in recent process technologies, and it will further increase while switching power becomes smaller when the technology is scaled down.

Different saving techniques have been used for the switching power and the leakage power. For example, clock gating, operand isolation, and dynamic voltage and frequency scaling (DVFS) have been used for switching power reduction, while multi-threshold voltages and power gating have been used for leakage power reduction. Therefore, combinations of these techniques are essential to reduce both the switching and leakage power.

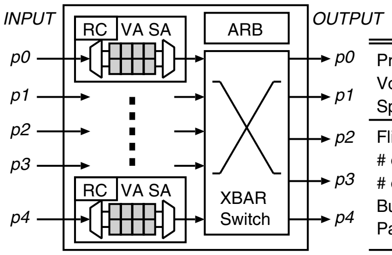
In this paper, we introduce the use of slow-silent virtual channels to reduce the switching power of on-chip networks while keeping the leakage power small. This proposal is based on a simple idea: adding virtual channels to a network improves the throughput until each link bandwidth is saturated. This enables us to reduce the switching power of on-chip networks by decreasing their operating frequency and supply voltage without degrading the throughput. However, adding virtual channels increases the leakage power of routers as well as the area due to their large buffers; so the runtime power gating is applied to individual virtual channels to eliminate this problem. Our claim is that the routers with extra virtual channels can reduce their power consumption if they are *slow* and *silent*.

The other contributions of this paper are detailed evaluations of slow-silent virtual channels in terms of switching and leakage power. These results will answer the question of how many virtual channels are needed to minimize the power consumption for a given traffic pattern.

The rest of this paper is organized as follows. Section 2 shows architecture of a typical on-chip router and analyzes its power consumption. Section 3 surveys the low-power techniques on microprocessors and NoCs. Section 4 introduces the slow-silent virtual channels and their sleep control method. Section 5 confirms that adding slow-silent virtual channels reduces the overall power consumption through evaluations, and Section 6 concludes this paper.

2 On-Chip Virtual-Channel Router

Prior to discussing low power techniques for on-chip routers, an architecture of a simple on-chip virtual-channel router is presented, and then its dynamic and static power consumption is analyzed with the method proposed in [2].



Router parameters	
Process	ASPLA 90nm
Voltage	1.0V
Speed	Max 500 MHz
Flit size	64-bit
# of ports	3, 4, 5, 6 (configurable)
# of VCs	1, 2, 3, 4 (configurable)
Buf size	4-flit for each VC
Packet len	4-flit data + 1-flit header

Figure 1. Router parameters and an example

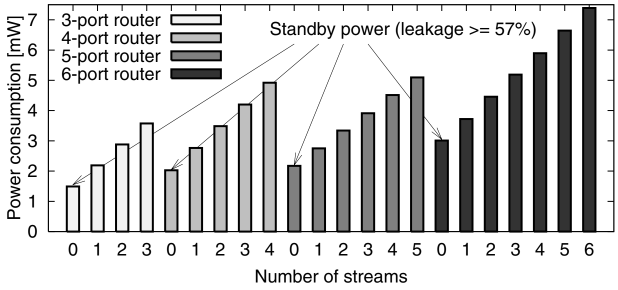


Figure 2. Active power at various workloads (3-6 ports; 4 VCs; 200MHz)

2.1 Router Architecture

For investigation on NoC architectures, we have implemented a wormhole router that has up to six physical channels and up to four virtual channels. We also developed an NoC generator that automatically connects the routers in arbitrary network topologies. The generated NoC is synthesized, placed, and routed with a 90nm standard cell library.

Figure 1 illustrates the router architecture used in this paper. This router consists of a crossbar switch (XBAR), an arbitration unit (ARB), and five input physical-channels (or ports), each of which has two virtual channels. Notice that the numbers of ports and virtual channels are configurable.

Each physical channel consists of a routing computation (RC) unit and up to four virtual channels, each of which has a FIFO buffer for storing four 64-bit flits. The RC unit in this design is very simple, because routing decisions are stored in the header flit prior to packet injection (i.e., source routing); thus routing tables that require register files for storing routing paths were not needed.

This is a typical input buffered router, which has buffers at only its input channels. These FIFO buffers can be implemented with either SRAMs or registers, depending on the depth of the buffers, not the width. We assume that buffers should be implemented with SRAM macros if their depths are more than 32. Otherwise buffers should simply be implemented with registers. Since the depth of the FIFO buffers in this design was only 4, the input buffers were implemented with small registers. As mentioned above, we used a small $p \times p$ crossbar and a simple RC unit with no routing tables. As a result, up to 67% of the total router area was used for the buffers when each port had four virtual channels.

The router architecture is fully pipelined. Although some 1- or 2-cycle routers have been developed by using some aggressive techniques[6][15], we selected a simple 3-cycle router architecture with no fancy techniques. Thus, our router transfers a header flit through three pipeline stages that consist of a routing computation (RC), a virtual channel and switch allocation (VSA), and a switch traversal (ST).

2.2 Power Analysis

To estimate the power consumption of the router mentioned previously, the following steps were performed: 1) a synthesis done by Synopsys Design Compiler, 2) a place and route, including a clock tree synthesis and buffer insertion, done by Synopsys Astro, 3) a post place-and-route simulation, done by Cadence Verilog-XL, to obtain the switching activity information of the router, and 4) a power analysis based on the switching activity done by Synopsys Power Compiler. A 90nm CMOS process with a core voltage of 1.0V was selected for this analysis. Clock gating and operand isolation were fully applied to the router to minimize its switching activity and dynamic power.

In step 3), the router was simulated at 200MHz to 500MHz with various fixed workloads (i.e., throughputs), in the same manner as in [2]. A packet stream is defined as intermittent injections of packets that utilize approximately 30% of the maximum link bandwidth of a single router link. Each header flit contains a fixed destination address, while the data flits contain random values as a payload. The number of packet streams injected into the router was changed so as to generate various workloads. In this experiment, up to n streams were applied to a router that has n physical channels (i.e., n ports), and the power consumption at each workload level was analyzed, where $3 \leq n \leq 6$.

Figure 2 shows the results on a router that has up to six physical channels, each of which has four virtual channels, in the case of 200MHz[1]. The router consumes more power as it processes more packet streams, in the following way:

$$P_{total} = P_{standby} + xP_{stream}, \qquad (1)$$

where x is the number of packet streams and P_{stream} is the dynamic power for processing a packet stream.

Notice that the router consumed a certain amount of power even with no traffic (i.e., $P_{standby}$). Figure 3 shows the breakdowns of the standby power for the 5-port router running at 200MHz and 500MHz. Leakage power consumes a substantial portion of the standby power. In particular, the percentage of the leakage power consumed in the virtual channels to the overall standby power is shown in the graph. As shown, the virtual channels consumed up to 49.4% of the standby power due to their FIFO buffers. The remaining power is the dynamic power consumed by the clock tree buffers and the latches inserted for the clock gating; so further reduction of the switching activity would be difficult.

[1] Although we obtained a lot of evaluation results from these experiments, only some of them are shown here to make our point.

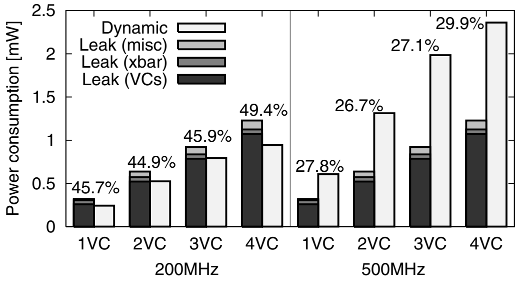


Figure 3. Breakdowns of the standby power (5 ports; 1-4 VCs; 200-500 MHz)

From the above discussion, we need to reduce P_{stream} while keeping $P_{standby}$ small. In Section 4, we will present a possible solution that meets these requirements by simply adding extra virtual channels running at a lower frequency, each of which can be power-gated if it is not used.

3 Low-Power Techniques

Various low-power techniques have been used for microprocessors and on-chip routers. In particular, clock gating and operand isolation are common techniques and they have already been applied to our router design. In this section, we survey the voltage and frequency scaling in Section 3.1, the runtime power gating in Section 3.2, and the dynamic channel and link shutdown of networks in Section 3.3.

3.1 Voltage and Frequency Scaling

The voltage and frequency scaling is a power saving technique that reduces the operating frequency and supply voltage according to the applied load. It has been applied to microprocessors[8][16][17], accelerators [12], and network links[19][21]. In [19], the frequency and the voltage of network links are dynamically adjusted based on the past utilization. In [21], the network link voltage is scaled down by distributing the traffic load using an adaptive routing. In [12], the supply voltage of a H.264/AVC decoder is adaptively changed between 1.00V and 0.55V.

The voltage and frequency scaling techniques can be classified into two schemes: *dynamic* and *static*. The frequency can be controlled by a PLL frequency divider, and the supply voltage can be adjusted by controlling an off-chip dc-dc converter[16]. They are adaptively adjusted in the dynamic scheme, while they are statically configured at the beginning of each application in the static scheme. The transition time of the clock rate and the supply voltage cannot be negligible in the dynamic scheme (e.g., 10,000 cycles[21] or 50μs[12]); so the frequent transitions sometimes overwhelm the benefits of dynamic scheme.

In this paper, the frequency and the voltage of on-chip routers are reduced by enhancing the number of virtual channels. They are adjusted per application basis to eliminate the frequent transitions of the clock frequency, but our technique can also be applied to finer-grain adjustments.

3.2 Runtime Power Gating

Power gating is a representative leakage-power reduction technique, which shuts off the power supply of idle blocks by turning off (or on) the power switches inserted between the VDD line and the blocks or between the GND line and the blocks. This concept has been applied to circuit blocks with various granularities, such as processor cores[11], execution units in a processor[10], and primitive gates[22]. In this paper, we focus on the execution unit level, since its granularity is close to a virtual channel in the on-chip router.

We need to understand both the negative and positive impacts of power gating when we use it. Actually, a state transition between the sleep and active modes incurs a performance penalty, and turning the power switches on or off dissipates an overhead energy, which means a short-term sleep rather increases the power consumption. In [10], an analytical model of the run-time power gating of execution units in a microprocessor is provided. The following three parameters quoted from [10] affect the performance and energy.

- T_{wakeup}: Number of cycles required to charge up the local voltage of a sleeping block. A delay for turning on its power switch is also lumped into the T_{wakeup} value.
- $T_{idledetect}$: Number of cycles required to detect an idle duration in an active block and decide to shut off the block. A delay for turning off its power switch is also lumped into the $T_{idledetect}$ value.
- $T_{breakeven}$: Number of sleep cycles at least required to compensate for the overhead energy to turn the power switch on and off.

The T_{wakeup} value affects the performance (e.g., packet throughput of routers), since a pipeline stall will occur if a new request suddenly comes to a sleeping block. Also, $T_{idledetect}$ shortens the sleep duration of blocks, since an idle block must stay in the active state for $T_{idledetect}$ cycles before it decides to go to the sleep mode.

A short-term sleep of less than $T_{breakeven}$ cycles cannot compensate for the energy overhead of driving a power switch, and the power consumption will be increased; thus the $T_{breakeven}$ value determines the benefits of power gating. The $T_{breakeven}$ value depends on various parameters, such as the sizes of a power switch and a decoupling capacitance. Reference [10] reports that $T_{breakeven} \approx 10$ based on the typical parameters of a recent microprocessor.

3.3 Dynamic Link/Channel Shutdown

Reference [4] proposes power-aware router buffers based on Drowsy and Gated V$_{dd}$ SRAMs to regulate their leakage power. Reference [20] provides a thorough discussion about power-aware networks whose links can be turned on and off, in terms of connectivity, routing, wake-up and sleep decisions, and router pipeline architecture.

These works are proposed for both off-chip and on-chip interconnects, and they assume to use relatively large buffers in their routers, compared to those in the simple on-chip wormhole routers[14]. In [4], the router buffers are constructed with SRAMs. As a sleep control policy for the buffer, a certain portion (i.e., window size) of the buffer is made active before it is accessed. By tuning the window size, the input ports can always provide enough buffer space for the arrival of packets, and the network performance will never be affected[4]. Our light-weight wormhole router uses a small 4-flit buffer for each virtual channel, unlike the routers that use SRAMs for their input and output buffers. Since the buffer depth is shorter than the window size in low-cost routers, the wake-up delay of the buffers directly affects the network performance if the links or channels are dynamically turned on and off.

4 Slow-Silent Virtual Channels

In this section, we introduce the slow-silent virtual channels to reduce the on-chip network's overall power by adding extra virtual channels running with lower clock frequencies. We assume that the clock frequency is controlled by the PLL divider, and the supply voltage is adjusted by the off-chip dc-dc converter. They are configured for each application, but finer-grain adjustment is also possible. The problem of this approach is the increase of leakage power due to the extra virtual channels; thus runtime power gating is applied to each virtual channel to alleviate this problem.

We first discuss the voltage and frequency scaling of virtual channels. Then we propose a sleep control method and routing strategy for virtual-channel level power gating.

4.1 Adding Slow Virtual Channels

The saturated throughput of a network is the data acceptance rate, in which the communication latency goes to infinity. Here, we simply express the saturated throughput of a given network with v virtual channels as Θ_v. The Θ_v value highly depends on the network topology, routing algorithm, network size, traffic pattern, and arbitration technique. In this paper, a network simulator is used to obtain the Θ_v.

Assume that a given network has v virtual channels and its performance, Θ_v, meets the requirements of the target application. When we add extra n virtual channels to the network, the performance improvement can be expressed as Θ_{v+n}/Θ_v. This means that the operating frequency of the network can be reduced by Θ_v/Θ_{v+n} without degrading the original throughput [2].

There is a relationship between the operating frequency and the supply voltage. The gate delay dependence on the supply voltage can be approximated by

$$T_{delay} \propto \frac{CV}{(V - V_{th})^{\alpha}}, \tag{2}$$

where T_{delay} is the gate delay, C is the capacitance being switched, V is the supply voltage, V_{th} is the threshold voltage, and α is the velocity saturation index in a short channel MOSFET[18] and is about 1.6 as reported in [12]. Therefore, adding extra virtual channels can reduce the supply voltage as well as the operating frequency.

The dynamic switching power can be expressed as

$$P_{sw} = a \cdot C \cdot f \cdot V^2, \tag{3}$$

where a is the switching activity, C is the capacitance, and f is the operating frequency. Although adding extra virtual channels increases the effective switched capacitance (i.e., aC), it can reduce the operating frequency and the supply voltage. Section 5 evaluates the switching power reduction of networks with up to four slow-silent virtual channels whose frequency is adjusted for a given application.

4.2 Silent Virtual Channels

In the previous section, we showed the probability of the switching power reduction with the addition of extra virtual channels. However, adding virtual channels proportionally increases the buffer area of the light-weight wormhole routers mentioned in Section 2. The leakage power consumption is proportional to the device area; thus adding extra n virtual channels to an original network that has v virtual channels increases the overall leakage power by $(v + n)/v$. Since the leakage power has already become a major component of the power consumption in on-chip routers, the leakage power of extra virtual channels may overwhelm the switching power reduction obtained by the voltage and frequency scaling.

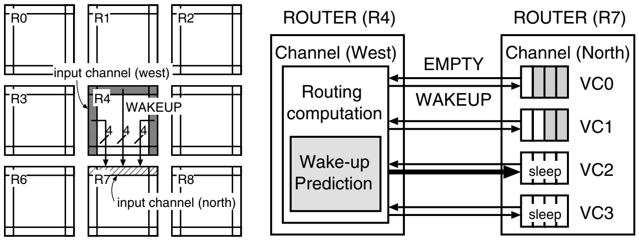
The runtime power gating of individual virtual channels can mitigate this problem. Ideally, only the virtual channels occupied by packets are active and consume leakage power when the runtime power gating is applied to each virtual channel. This means that the leakage power consumption is proportional to the data acceptance rate (i.e., throughput), rather than the number of virtual channels. However, the above discussion assumes an ideal case in which the negative impacts of power gating are ignored.

4.3 Sleep Control for Early Wake-Up

Delay to activate a sleeping virtual channel (T_{wakeup}) affects the performance of the network, since a pipeline stall will occur if a new request suddenly comes to the sleeping virtual channel. In this section, we propose a sleep control method that detects the arrival of the packets three cycles ahead, so as to mitigate the negative impacts of T_{wakeup}.

Here, a "sender" refers an input physical channel that transmits a packet and a "receiver" refers an input physical channel that receives the packet. Each virtual channel in a receiver monitors the "wakeup" signal, which indicates that new packets (or requests) are approaching to the virtual channel. These wakeup signals are controlled by senders directly connected to the receiver. Figure 4(a) shows the wakeup signals that control the north channel of router 7 from three channels of the neighboring router (i.e., west, north, and east channels of router 4 are senders). Figure 4(b) shows the detail of wakeup signals between the west

[2] It is also possible to consider the throughput value at a certain communication latency (e.g., 200 cycles) instead of the saturated throughput.



(a) Network overview (b) West ch. of R4 ↦ North ch. of R7

Figure 4. Wake-up signals for north channel of R7 (R denotes a router)

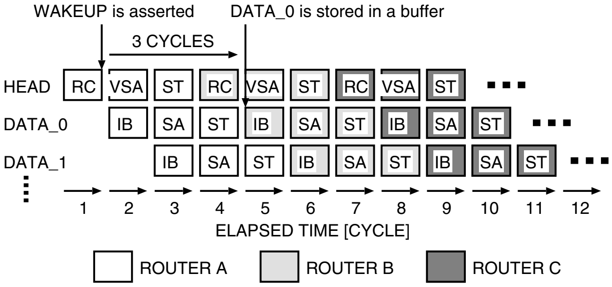


Figure 5. Router pipeline with early wake-up

channel of router 4 and the north channel of router 7. The wake-up control procedures of the receiver and the sender are described separately.

Receiver The receiver uses the wakeup signals to decide which virtual channel in the receiver should be activated. That is, the receiver activates a sleeping virtual channel when the wakeup signal to the sleeping virtual channel is asserted by the senders. The receiver also uses the wakeup signals to decide which virtual channel should sleep. The receiver checks the wakeup signals after forwarding a packet, and if the wakeup signal to the empty virtual channel is de-asserted, the empty virtual channel will be power-gated. The performance penalty due to a wake-up delay depends on how early new requests can be detected.

Sender To reduce the wake-up delay, the sender notifies the directly-connected receiver that a packet will be arriving at the receiver a few cycles later. Figure 5 shows the router pipeline, in which a packet consisting of a header flit and several data flits is transferred from router A to router C. At cycle 1 in Figure 5, the RC stage of router A detects which input physical-channel of router B is going to be used.

In our design, the RC stage of the sender predicts which virtual channel is going to be used in the receiver. The RC stage of a sender performs the following three steps: 1) the output channel of the incoming packet is computed (i.e., receiver is selected); 2) a virtual channel to be used in the receiver is predicted; and 3) the wakeup signal to the predicted virtual channel is asserted to activate the virtual channel. Note that step 1) is the normal RC operation while steps 2) and 3) are newly introduced for our router design.

There are some ways to predict it, but we employ a very simple one, which selects a single empty virtual-channel of the least VC number in the receiver. In Figure 4(b), VC2 is the empty virtual channel of the least VC number in the receiver; so the sender selects VC2 and asserts the wakeup signal for VC2 in the receiver. We implemented the prediction unit in the RC stage of our router. The prediction unit was implemented with simple combinational logic and the critical path of the router was not affected.

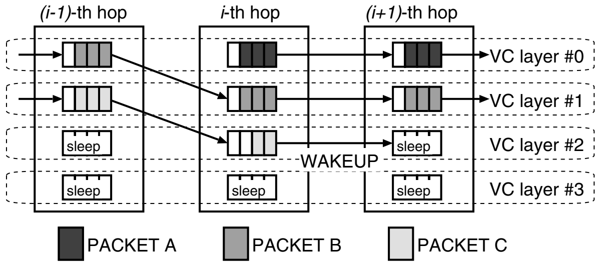


Figure 6. A routing example with 4 VC layers

As a flow control, the sender ST stage monitors the status (e.g., active or sleep) of each virtual channel in the receiver, and it forwards data flits to one of virtual channels in the receiver if the virtual channel is ready to receive. As shown in Figure 5, at cycle 5, the first data flit (i.e., DATA_0) is stored in a virtual channel in router B. Since the virtual channel must be activated before cycle 5, there is a three-cycle margin after router A asserts the wakeup signal.

If the prediction is right, the performance penalty is mitigated, or it is removed when $T_{wakeup} \leq 3$. Otherwise, incoming packets must wait T_{wakeup} cycles until the allocated virtual channel in the receiver wakes up. Notice that the performance penalty induced by the miss predictions is small, because a miss prediction occurs only when more than two senders select the same virtual channel in the same receiver at the same time[3].

4.4 Routing Design

Since active virtual channels consume a standby power, they should be gradually activated according to the traffic load of the network. We designed a packet routing for networks with multiple slow-silent virtual channels, as follows.

All packets are injected into the network via the virtual-channel number 0 (VC0). Then the packet increments its virtual-channel number whenever it conflicts with other packets on the original virtual channel. Thus only the VC0 is activated at the low traffic load, while the other virtual channels are additionally activated as the traffic load increases (Figure 6). This enables us to achieve a high peak performance with the least standby power of routers. Notice that all virtual channel layers except for the bottom layer (i.e., VC layer #3) can employ arbitrary routing algorithms as far as the bottom layer guarantees the deadlock-freedom by itself. This property is known as Duato's protocol [7] and it provides a flexibility for the routing designs.

[3] If the three-cycle margin is not enough, it is possible to employ a look-ahead based sleep control method that detects the arrival of packets five cycles ahead[14]. However, the method introduces long wakeup signals that cover the twice longer length than normal links between routers cover.

5 Evaluations

In this section, we demonstrate that adding slow-silent virtual channels to a network can reduce its switching power while keeping its leakage power small.

Assuming that the throughput of a network with no virtual channels (i.e., 1-VC network) meets the requirements of the target application, the voltage and frequency scaling technique is applied to 2-VC, 3-VC, and 4-VC networks. As preliminary evaluations, Section 5.1 and 5.2 evaluate the original throughput and power consumption of these networks with various application traces, respectively. Based on these results, Section 5.3 calculates how much operating frequency and supply voltage can be scaled down. Section 5.4 evaluates the overhead energy and the leakage power reduction of the runtime power gating. Finally, Section 5.5 evaluates the overall power reduction when both the voltage and frequency scaling technique and the runtime power gating technique are applied to these networks.

5.1 Original Performance

We first introduce the simulation environment used in this experiment, and then we evaluate the network throughput without frequency scaling (i.e., original performance).

Simulation Environment A flit-level network simulator written in C++ was used for this experiment. A simple wormhole router mentioned in Section 2.1 was used as the switching element in the simulator. That is, each router had up to four virtual channels, it had three pipeline stages, and each packet consisted of five flits including one flit header (see Figure 1). The network topology used in this simulation was an 8×8 two-dimensional mesh, and dimension-order routing (DOR)[6] was used for each virtual-channel layer. The simulation time was set to at least 200,000 cycles, and the first 1,000 cycles were ignored to avoid distortions due to the startup transient.

As for the traffic patterns, we used five application traces captured from NAS Parallel Benchmark (NPB)[1] as well as [13] and [14]. NPB consists of typical numerical parallel application programs described with MPI library, and it includes various traffic patterns, such as all-to-all and stream fork/join. To conduct evaluations using wide range of traffic patterns, we selected the following five programs: Block Tridiagonal solver (BT), Scalar Pentadiagonal solver (SP), Conjugate Gradient (CG), Multi-Grid solver (MG), and large Integer Sort (IS). The class of problems was set to "W", and the numbers of tasks to 64. In addition to these programs, we used uniform random traffic as a baseline for comparison. For each traffic pattern, we evaluated its throughputs at various workloads by linearly changing time span between packet transfers (i.e., time compression).

Simulation Results Figure 7(a) shows the throughput (accepted traffic) versus the latency using 64-core uniform traffic on 1-VC, 2-VC, 3-VC, and 4-VC networks. The average hop count is shown in the caption of the graph. The performance increases as the number of virtual channels increases. The performance improvement from 1-VC to 2-VC is the largest, while that from 3-VC to 4-VC is not so large. A similar result can be seen in IS.W traffic, which contains many all-to-all communications (Figure 7(f)). In the BT.W and SP.W traces, the performance improvement from 2-VC to more is saturated, since their communication patterns contain many neighboring communications insusceptible to the head-of-line blockings (Figure 7(b) and 7(c)).

Table 1. Original throughput [Mflit/sec/core]

	1VC	2VC	3VC	4VC
uniform	56.08	92.68	116.9	123.2
BT.W	92.54	131.9	133.0	132.0
SP.W	88.08	120.3	126.6	125.8
CG.W	70.20	111.3	124.2	123.3
MG.W	92.37	132.6	132.5	131.5
IS.W	57.91	102.6	118.6	125.6

Table 1 summarizes the saturated throughput. Note that we assume that the 1-VC, 2-VC, 3-VC, and 4-VC networks are running at 500.0MHz, 498.8MHz, 497.7MHz, and 493.8MHz, respectively. These frequency values were obtained from the placed and routed design of each network with the same design constraints.

5.2 Original Power Consumption

We introduce the energy model of the NoCs used in this experiment, and then we evaluate the power consumption without voltage scaling (i.e., original power consumption).

Active Power Model Our energy model takes into consideration the active and the standby power separately. The average energy consumption needed to transmit a single flit from a source to a destination can be estimated as

$$E_{flit} = wH_{ave}E_{link} + w(H_{ave} + 1)E_{sw}, \qquad (4)$$

where w is the flit-width, H_{ave} is the average hop count, E_{sw} is the average energy to switch the 1-bit data inside a router, and E_{link} is the 1-bit energy consumed in a link.

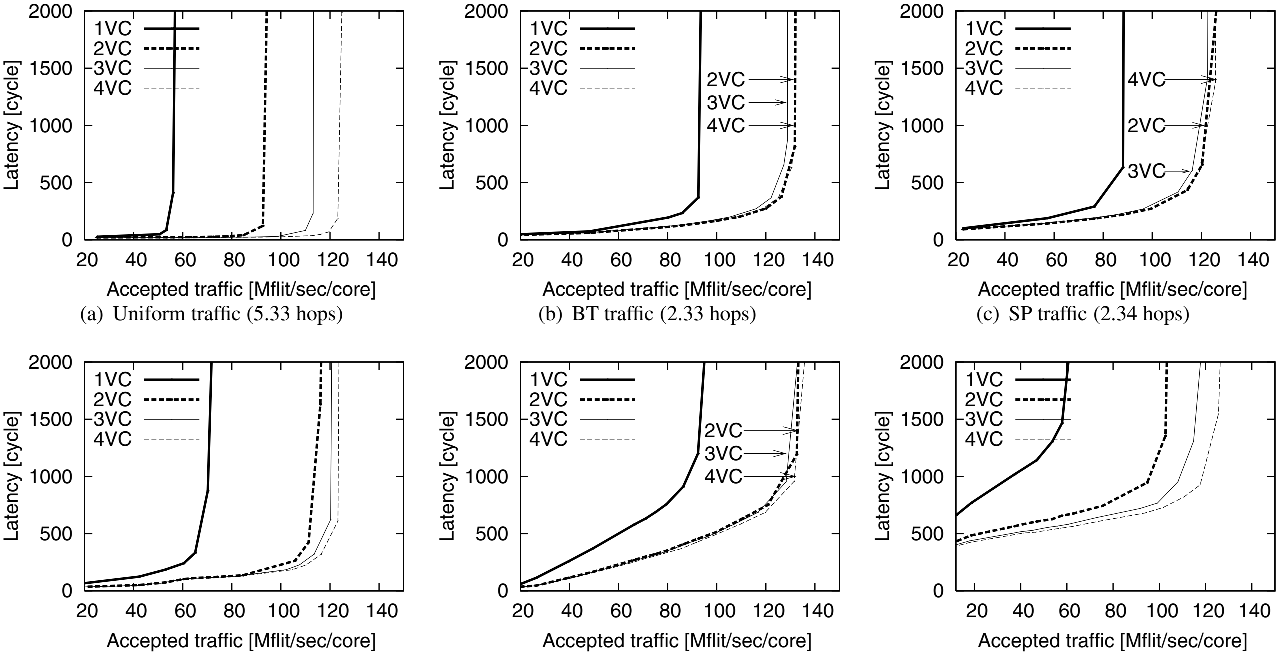
We used Synopsys Power Compiler to extract the E_{sw} of the router placed and routed with the 90nm technology. The switching activity of the running router was captured through the post place-and-route simulation of the router operating at 500MHz with a 1.0V supply voltage. The gate-level power analysis based on this switching activity shows that E_{sw} is 0.144pJ for 1-VC, 0.153pJ for 2-VC, 0.154pJ for 3-VC, and 0.156pJ for 4-VC network, respectively.

E_{link} can be calculated as

$$E_{link} = dV^2C_{wire}/2, \qquad (5)$$

where d is the 1-hop distance (in millimeters), V is the supply voltage, and C_{wire} is the wire capacitance per millimeter. C_{wire} can be estimated using the method proposed in [9], and is 300fF/mm in the case of a semi-global interconnect in the 90nm CMOS technology. For instance, E_{link} is 0.150pJ when the 1-hop distance is 1mm on average.

We assumed 64-bit 64-core networks placed in an 8mm × 8mm chip; thus the size of each tile is 1mm × 1mm.



(a) Uniform traffic (5.33 hops)
(b) BT traffic (2.33 hops)
(c) SP traffic (2.34 hops)
(d) CG traffic (2.93 hops)
(e) MG traffic (2.94 hops)
(f) IS traffic (5.14 hops)

Figure 7. Original performance

The 1-hop distance d was set to 0.7mm, since a router consumes a 0.3mm × 0.3mm area of the tile. Then we derived E_{flit} using Equation 4 with the various parameters mentioned above.

Standby Power Model The standby power includes: 1) the leakage power of routers, 2) the dynamic power of clock tree buffers, and 3) the dynamic power of the latches inserted for the clock gating. They were estimated based on a post place-and-route simulation with no traffic load.

Evaluation Results Figure 8(a) shows the applied traffic load versus the overall power consumption (left scale) and the leakage power (right scale) with uniform traffic. The overall power consumption at zero traffic load is the standby power. The total power increases as the traffic load increases, while the leakage is constant. The leakage power is proportional to the network logic area; thus a 4-VC network consumes the largest standby power. Similar results can be seen in the NPB traces. These results show that adding extra virtual channels without voltage scaling and runtime power-gating techniques increases the power consumption. We will present the effect of the voltage scaling and the runtime power gating in the following sections.

5.3 Voltage and Frequency Scaling

Assuming that the throughput of a 1-VC network running at 500MHz meets the requirements of the target application, the frequencies of the other networks can be scaled down, as shown in Table 2. The more performance is improved, the more frequency can be reduced. Therefore, the core voltages of networks with multiple virtual channels can be reduced based on Equation 2, and the results are shown in Table 3. In Section 5.5, we will show the final power consumption with scaled frequencies and core voltages.

Table 2. Scaled operating frequency [MHz]

	1VC	2VC	3VC	4VC
uniform	500.0	301.8	238.8	224.8
BT.W	500.0	350.1	346.2	346.1
SP.W	500.0	365.1	346.3	345.9
CG.W	500.0	314.6	281.3	281.3
MG.W	500.0	347.4	346.9	346.8
IS.W	500.0	281.5	243.1	227.8

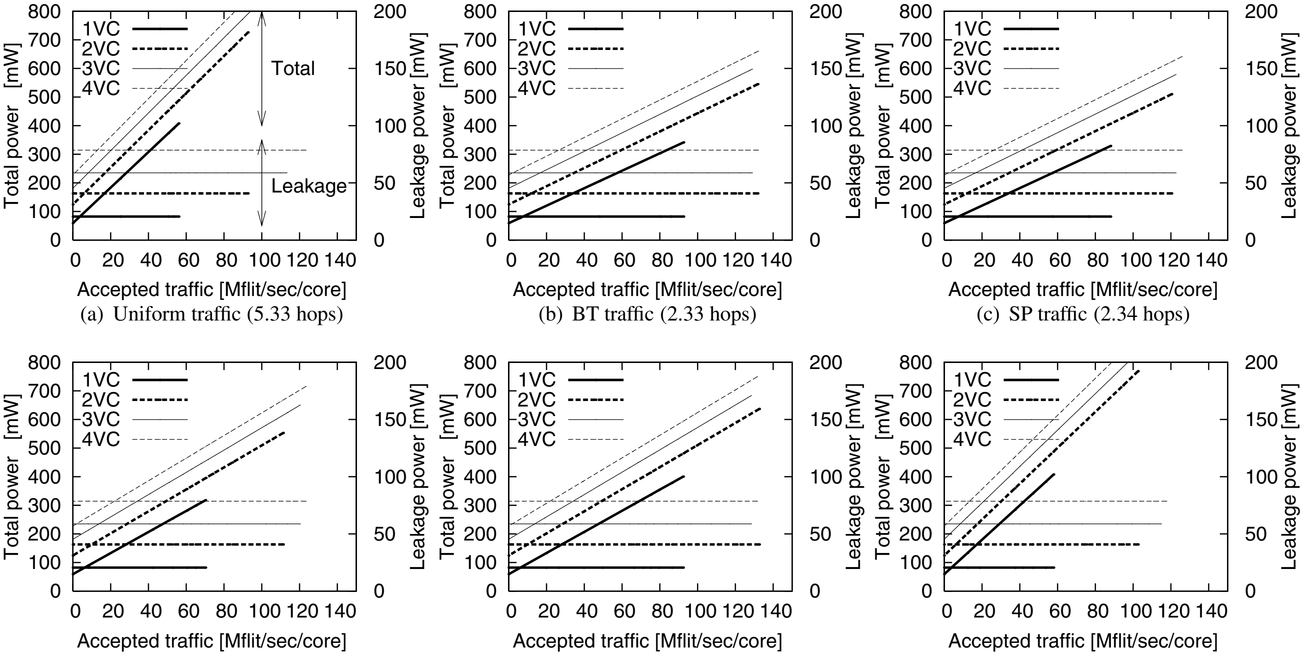
Table 3. Scaled supply voltage [V]

	1VC	2VC	3VC	4VC
uniform	1.00	0.77	0.70	0.68
BT.W	1.00	0.82	0.82	0.82
SP.W	1.00	0.84	0.82	0.82
CG.W	1.00	0.78	0.74	0.75
MG.W	1.00	0.81	0.82	0.82
IS.W	1.00	0.74	0.70	0.69

5.4 Runtime Power Gating

We introduce the leakage power model of the power gating, and then we evaluate the overhead energy and the leakage power reduction of the power-gated virtual channels.

The leakage power can be reduced by turning off the power switch of the circuit block if the sleep duration is longer than $T_{breakeven}$. Otherwise, the overhead energy to



(a) Uniform traffic (5.33 hops)

(b) BT traffic (2.33 hops)

(c) SP traffic (2.34 hops)

(d) CG traffic (2.93 hops)

(e) MG traffic (2.94 hops)

(f) IS traffic (5.14 hops)

Figure 8. Original power consumption (w/o voltage and frequency scaling; w/o power gating)

Table 4. Energy parameters of virtual channel

		200MHz	500MHz
V	supply voltage	1.0V	1.0V
E_{leak}	static energy	0.052mW	0.052mW
E_{sw}	dynamic energy	0.078mW	0.194mW
L	$= E_{leak}/E_{sw}$	0.67	0.27
a	switching factor	0.12	0.12

turn the power switch on and off would overwhelm the benefit of the power gating. Assuming that the overhead energy is lumped into the leakage power, we modeled the leakage power reduction of the power-gated virtual channel, in the same manner as proposed in [10].

The router was simulated at 200MHz to 500MHz to analyze the power consumption of each virtual channel. Table 4 shows the energy parameters of the router channel, such as the dynamic power, leakage power, and switching activity. These parameters were used in the following leakage power model to estimate that with runtime power gating.

As reported in [10], the total energy saved over N cycles (denoted as E^N_{saved}) can be calculated as follows:

$$E^N_{saved} = E_{leak}\frac{\text{DIBL}}{mV_t}\frac{N^2}{2}\frac{aLV}{2(\frac{1}{2}+\frac{C_D}{C_S})}, \qquad (6)$$

where DIBL is the drain-induced barrier lowering, which is typically close to 0.1, $V_t = kT/q \approx 25$mV is the thermal voltage, and $m \approx 1.3$ [10]. Also, W_H is the ratio of the total area of the power switch to the area of the target block (i.e., a virtual channel), C_S is the total switching capacitance of the block, and C_D is the total capacitance at the local power supply including the power switch. Typically, W_H is 0.1, and C_D/C_S is 0.5 as reported in [10].

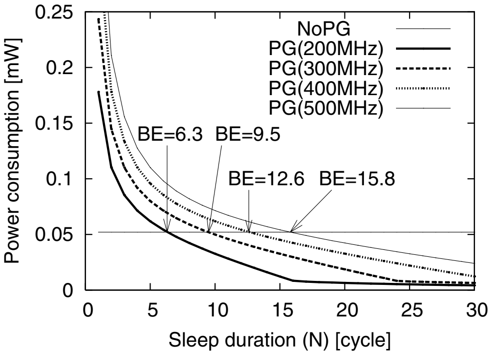


Figure 9. Leakage power vs. sleep duration

As reported in [10], the energy overhead to turn the power switch (denoted as $E_{overhead}$) can be calculated as follows:

$$E_{overhead} \approx 2\frac{W_H}{a}E_{leak}. \qquad (7)$$

Based on Equation 6 and 7, we calculated the average leakage power consumed in a virtual channel during an N-cycle sleep, where $0 < N < 100$. Again, the average leakage power consumption in this experiment includes the dynamic energy overhead to turn the power switch.

The results are shown in Figure 9. “NoPG” stands for the leakage power of the virtual channel without power gating, while “PG” shows the result of the power-gated channel at each operating frequency. As shown in the graph, the average leakage power of the power-gated channel is decreased as the sleep duration N gets longer. Since the leakage power of NoPG is 0.052mW, we can estimate that the $T_{breakeven}$ is 6.3 cycles for PG(200MHz), 9.5 cycles for PG(300MHz), 12.6 cycles for PG(400MHz), and 15.8 cycles for PG(500MHz).

5.5 Final Power Consumption

Here, we evaluate the overall power reduction when both the voltage/frequency scaling and the runtime power gating are applied to networks with multiple virtual channels.

Simulation Environment The energy model used in this section is the same as that in Section 5.2, except for the leakage power modeling. The leakage power was estimated as follows. We first performed the same network simulations with the runtime power gating of virtual channels, assuming that the T_{wakeup} and $T_{idledetect}$ were two and four cycles [14], respectively. From these simulation results, we obtained the length (i.e., number of cycles) of every sleep in every virtual channel during the total execution time. Then, we calculated the average power consumption for each sleep period according to its length, based on the "Leakage power vs. sleep duration" graph shown in Figure 9. That is, a sleep longer than $T_{breakeven}$ contributes for reducing the average power consumption, while a sleep shorter than $T_{breakeven}$ adversely eats up additional power.

Evaluation Results Figure 10 shows the final results. Just as in Figure 8, the left scale shows the overall power and the right scale shows the leakage power. As shown in these graphs, the overall power consumption is drastically reduced by the voltage and frequency scaling compared to the original power consumption (Figure 8). When the uniform traffic is injected into the 4-VC network at 56 million flits per second per core, the original network consumes 598mW (Figure 8(a)) while the network with voltage and frequency scaling consumes 250mW (Figure 10(a)); thus up to 58.2% of the total power was saved in this case.

Notice that the wakeup delay of a sleeping virtual channel did not affect the performance by the sleep control method that detects the arrival of packets three cycles ahead, because the T_{wakeup} was set to two cycles. In addition, the virtual channel allocation was packed into the RC stage of routers in this simulation for the baseline evaluation; thus the miss predictions did not happened.

The leakage power is also greatly reduced. Although the leakage power consumption without runtime power gating is constant (Figure 8), that with power gating consumes only the necessary leakage power (Figure 10). In the 4-VC network with uniform traffic, the leakage power of the original network is 79mW, while that with runtime power gating increases from 12mW to 46mW as the traffic load increases; thus the leakage power was saved by 40.9%-84.9%.

In the cases of uniform and IS.W (Figure 10(a) and 10(f)), 3-VC and 4-VC networks achieve the best energy efficiency, because their performance can be well improved by adding more than two virtual channels. In these traffic patterns, the 4-VC network consumes 37.8%-40.7% less energy than the 1-VC network at the peak throughput. In the cases of BT.W and SP.W, on the other hand, the 2-VC network achieves the best energy efficiency, followed by 3-VC, 4-VC, and 1-VC networks. This is because the performance improvement from 1-VC to 2-VC is the largest but those from 2-VC to more are not so crucial, as their traffic patterns contain a lot of neighboring communications. Notice that the energy efficiencies of 3-VC and 4-VC are better than that of 1-VC in all cases; thus adding extra virtual channels can reduce the overall power consumption if they are *slow* and *silent*.

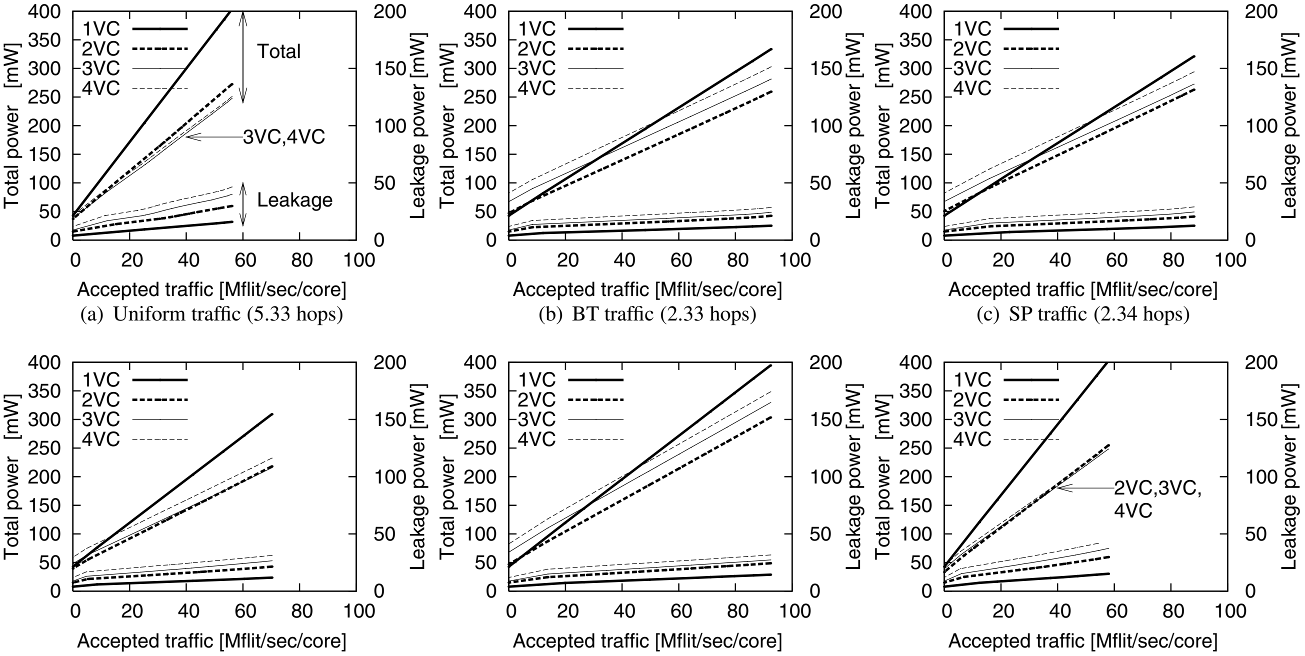
6 Conclusions

The combinations of the switching and leakage power reduction techniques are essential to reduce the overall power consumption of on-chip networks. Adding the slow-silent virtual channels proposed here is a low-power technique that scales down the operating frequency and the supply voltage of routers and alleviates the leakage power by the runtime power gating of individual virtual channels. We also proposed their routing strategy and sleep control method that detects the arrival of packets three cycles ahead.

To evaluate the performance and the power consumption of slow-silent virtual channels, we developed a cycle-accurate network simulator that can model the leakage power reduction by the runtime power gating, based on the detailed design of on-chip router with a 90nm technology. The evaluation results show that the 4-VC network with the voltage and frequency scaling and the runtime power gating reduces the leakage power by up to 84.9% and the overall power by up to 58.2% compared with the original network. We also investigated how many virtual channels are needed to minimize the power consumption for a given traffic pattern. In all-to-all communications, the 3-VC and 4-VC networks achieve the best energy efficiency and they reduce up to 40.7% energy compared to the 1-VC network, while the 2-VC network is better than the other networks in the cases of neighboring communications due to the less performance improvements from 2-VC to more virtual channels.

The concept of slow-silent virtual channels is versatile, and can be applied to other network designs. We are planning to apply this concept to fat tree topologies consisting of multiple trees, each of which can be individually power-gated according to the traffic load. That is, adding extra *slow* and *silent* trees to the original fat tree would reduce the overall power consumption. We will carefully consider such possibilities as a future work.

Acknowledgments A part of this work was supported by Toshiba Semiconductor Company, NII, and JST CREST. We would like to thank VLSI Design and Education Center (VDEC) and Kyoto University for a design flow of ASPLA/STARC 90nm CMOS process. Also, we would like to thank the reviewers for their valuable comments that improved our router design and this paper.



(a) Uniform traffic (5.33 hops)
(b) BT traffic (2.33 hops)
(c) SP traffic (2.34 hops)
(d) CG traffic (2.93 hops)
(e) MG traffic (2.94 hops)
(f) IS traffic (5.14 hops)

Figure 10. Final power consumption (w/ voltage and frequency scaling; w/ power gating)

References

[1] D. Bailey, T. Harris, W. Saphir, R. Wijngaart, A. Woo, and M. Yarrow. The NAS Parallel Benchmarks 2.0. *NAS Technical Report, NAS-95-020*, Dec. 1995.

[2] A. Banerjee, R. Mullins, and S. Moore. A Power and Energy Exploration of Network-on-Chip Architectures. *Proceedings of the International Symposium on Networks-on-Chip*, pages 163–172, May 2007.

[3] L. Benini and G. D. Micheli. *Networks on Chips: Technology And Tools*. Morgan Kaufmann, 2006.

[4] X. Chen and L.-S. Peh. Leakage Power Modeling and Optimization in Interconnection Networks. *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 90–95, Aug. 2003.

[5] W. J. Dally and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. *Proceedings of the Design Automation Conference*, pages 684–689, June 2001.

[6] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.

[7] J. Duato. A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems*, 4(12):1320–1331, Dec. 1993.

[8] S. Herbert and D. Marculescu. Analysis of Dynamic Voltage / Frequency Scaling in Chip-Multiprocessors. *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 38–43, Aug. 2007.

[9] R. Ho, K. W. Mai, and M. A. Horowitz. The Future of Wires. *Proceedings of the IEEE*, 89(4):490–504, Apr. 2001.

[10] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose. Microarchitectural Techniques for Power Gating of Execution Units. *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 32–37, Aug. 2004.

[11] M. Ishikawa et al. A 4500 MIPS/W, 86μA Resume-Standby, 11μA Ultra-Standby Application Processor for 3G Cellular Phones. *IEICE Transactions on Electronics*, E88-C(4):528–535, Apr. 2005.

[12] K. Kawakami, J. Takemura, M. Kuroda, H. Kawaguchi, and M. Yoshimoto. A 50% Power Reduction in H.264/AVC HDTV Video Decoder LSI by Dynamic Voltage Scaling in Elastic Pipeline. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, (12):3642–3651, Dec. 2006.

[13] H. Matsutani, M. Koibuchi, and H. Amano. Performance, Cost, and Energy Evaluation of Fat H-Tree: A Cost-Efficient Tree-Based On-Chip Network. *Proceedings of the International Parallel and Distributed Processing Symposium*, Mar. 2007.

[14] H. Matsutani, M. Koibuchi, D. Wang, and H. Amano. Run-Time Power Gating of On-Chip Routers Using Look-Ahead Routing. *Proceedings of the Asia and South Pacific Design Automation Conference*, Jan. 2008. (to appear).

[15] R. Mullins, A. West, and S. Moore. Low-Latency Virtual-Channel Routers for On-Chip Networks. *Proceedings of the International Symposium on Computer Architecture*, pages 188–197, June 2004.

[16] M. Nakai et al. Dynamic Voltage and Frequency Management for a Low-Power Embedded Microprocessor. *IEEE Journal of Solid-State Circuits*, 40(1):28–35, Jan. 2005.

[17] K. Nowka et al. A 0.9V to 1.95V Dynamic Voltage-Scalable and Frequency-Scalable 32b PowerPC Processor. *Proceedings of the International Solid-State Circuits Conference*, pages 340–341, Feb. 2002.

[18] T. Sakurai and A. R. Newton. Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas. *IEEE Journal of Solid-State Circuits*, 25(2):584–594, Apr. 1990.

[19] L. Shang, L.-S. Peh, and N. K. Jha. Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks. *Proceedings of the International Symposium on High-Performance Computer Architecture*, pages 79–90, Jan. 2003.

[20] V. Soteriou and L.-S. Peh. Exploring the Design Space of Self-Regulating Power-Aware On/Off Interconnection Networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(3):393–408, Mar. 2007.

[21] J. M. Stine and N. P. Carter. Comparing Adaptive Routing and Dynamic Voltage Scaling for Link Power Reduction. *IEEE Computer Architecture Letters*, 3(1):14–17, Jan. 2004.

[22] K. Usami and N. Ohkubo. A Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signals. *Proceedings of the International Conference on Computer Design*, Oct. 2006.

[23] S. Vangal et al. An 80-Tile 1.28TFLOPS Network-on-Chip in 65nm CMOS. *Proceedings of the International Solid-State Circuits Conference*, Feb. 2007.