

FinFET-based Dynamic Power Management of On-chip Interconnection Networks through Adaptive Back-gate Biasing

Chun-Yi Lee and Niraj K. Jha

Dept. of Electrical Engineering, Princeton University, Princeton, NJ 08544

{chunlee, jha}@princeton.edu

Abstract—On-chip interconnection networks are fast becoming significant power-consumers in high-performance chip multiprocessors (CMPs). Increased power consumption leads to more heat, adversely degrades system reliability, and may increase the cost of cooling IC packages. This situation becomes even worse as bulk CMOS scales further into the nanometer regime because of excessive leakage power due to short-channel effects. In this paper, we explore the use of FinFETs, which are promising substitutes for bulk CMOS at the 32nm node and beyond, to design on-chip network routers. We present a detailed design of a variable pipeline stage router (VPSR) targeted at FinFET technology. We employ a dynamic power management scheme, which we call adaptive back-gate biasing (ABGB), for FinFET implementations. We evaluate VPSR and ABGB on a simulation platform specifically designed for power and performance simulations for FinFET-based interconnection networks. The results show that VPSR is able to successfully adapt its power consumption to incoming traffic, with a resultant 20% reduction in power at almost no impact on latency.

Keywords – FinFETs, GARNET, interconnection network, ORION, VPSR, voltage generator.

I. INTRODUCTION

A continuous increase in the number of processor tiles and demand for high-bandwidth data communication in CMPs have led to a significant increase in the power consumed by on-chip packet-switched interconnection networks. A number of CMP architectures have been proposed by both academia [1]–[3] and industry [4], [5], and some of them have been shown to dissipate a lot of power in their on-chip network fabric. For instance, in the Alpha 21364 processor, the integrated router and links consume 20% of its total power [21]. The power from routers and links in Intel's teraflop processor is about 28% of the tile power [5]. The MIT Raw on-chip network consumes up to 36% of the total chip power [6]. High power consumption increases temperature, leading to hot spots, and hence degrades performance and reliability. Moreover, high power consumption increases the cost of cooling IC packages. Hence, dynamic power management (DPM) is urgently required for interconnection networks to meet their constrained power budgets [7], [8].

Another significant challenge we face is the increasing role played by leakage power in the overall CMP power consumption, not just in idle mode, but also in active mode. The active-mode leakage power in CMOS circuits was estimated to be as much as 40% of the total power consumption in the 70nm technology node [9]. In the 32nm FinFET technology, the active-mode leakage power is reported to be 78% [10]. As transistors scale down to deep sub-

micron technology nodes, leakage is becoming the dominant part of power consumption [11]. Moreover, various scaling obstacles are faced by bulk CMOS, such as short-channel effects (SCEs), process variations, dopant fluctuations, etc. Although circuit-level techniques like adaptive body-biasing (ABB) [17] has been proposed to reduce leakage, it has been reported in [12] that the role of well/body bias in threshold voltage modulation is becoming less effective as CMOS scales down. Therefore, underlying transistor-level solutions are needed to overcome the above problems.

This paper explores the use of FinFETs, which have emerged as promising substitutes for bulk CMOS at the 32nm technology node and beyond. FinFETs are reported to have much shorter delay and less power consumption compared to traditional CMOS devices at the 32nm technology node [13]. Moreover, the use of lightly doped channel makes FinFETs more resistant to process variations. Independent control of the two transistor gates in FinFETs enables a number of novel design styles. FinFETs also occupy less area because fin height determines effective channel width.

In this paper, we first present a detailed design of VPSR. VPSR is especially well-suited to FinFET-based interconnection networks, with the aim of dynamically managing the power consumption while maintaining performance. The DPM method used is ABGB. ABGB dynamically controls the back-gate bias voltages of FinFETs in order to produce various power and delay characteristics. Voltage generators are incorporated into VPSR to enable run-time exploitation of the ABGB concept. A flow control mechanism is developed for VPSR to adjust its performance to meet the current traffic requirement. Also, a power and performance simulation platform is developed specifically for FinFET-based interconnection networks to make VPSR simulation possible. Our experiments show that VPSR is able to reduce 20% power on an average at runtime with little impact on network latency.

The remainder of this paper is organized as follows. Section II introduces background material related to this work. Section III walks through the basic idea of VPSR and the design decisions made. Section IV discusses the DPM scheme used in VPSR in detail. Section V explains the power and performance simulation platform for FinFETs. Section VI explores the power breakdown and network latency of FinFET-based interconnection networks. Section VII concludes.

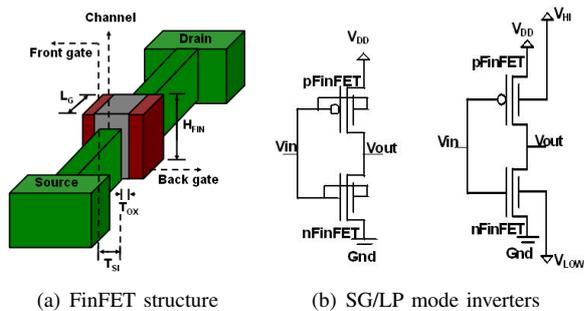


Fig. 1. FinFET structure and operation modes

II. BACKGROUND

In this section, we discuss the FinFET structure and its operation modes first. This is crucial to the understanding of ABGB. To understand how the proposed VPSR works, we briefly describe the microarchitecture and pipeline stages of a typical on-chip router.

A. FinFET Structure and Modes

Fig. 1(a) shows the FinFET structure. A fin is used as the channel body. The gate length (L_g) is equal to the fin length. The gate width is quantized by the number of fins, and is obtained by multiplying the fin count with twice the fin height (H_{FIN}). The front and back gates are on opposite sides of the fin, and can either be shorted or independently biased to different voltages. Shorted-gate (SG) and low-power (LP) modes are two possible ways to implement FinFET logic gates [14].

Fig. 1(b) illustrates SG- and LP-mode inverters. In the SG mode, the front and back gates are tied together. In the LP mode, the back gate of the pFinFET (nFinFET) is reverse-biased to a separate V_{HI} (V_{LOW}). The threshold voltage of the front gate varies linearly with the back-gate bias voltage, which can be adjusted to the desired value [14]. Hence, the leakage current of LP-mode gates can be significantly reduced by reverse-biasing their back gates. A higher reverse bias results in much smaller current, but increased gate delay due to the higher threshold voltage. It is reported in [15] that at 105°C , the leakage power of a normally-biased ($V_{HI} = 1.0\text{V}$ and $V_{LOW} = 0\text{V}$) LP-mode router can be as high as 7X as compared to a reverse-biased ($V_{HI} = 1.2\text{V}$ and $V_{LOW} = -0.2\text{V}$) router. Therefore, an effective and efficient DPM scheme is required to exploit this concept, which is discussed later.

B. On-chip Router

The function of a router is to route flits from the input ports to their requested output ports. Flits injected into the network must compete for router pipeline resources on a hop-by-hop basis along the path from their source nodes to destination nodes. Therefore, the number of router pipeline stages dominates both flit latency and router power consumption.

Fig. 2 shows the microarchitecture and the pipeline stages of a conventional five-stage router. As a flit arrives at the input port of the router, it is first written into the input buffer in the buffer write (BW) stage. In parallel, a route calculation (RC) unit determines the output port where the flit is to be

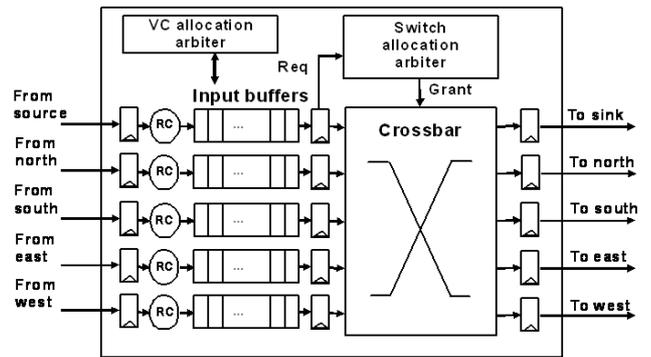


Fig. 2. Router microarchitecture and pipeline stages

forwarded. If the flit is the header of the packet, it goes through the virtual channel allocation (VA) stage and tries to acquire a free virtual channel (VC). Next, in the switch allocation (SA) stage, the switch allocator determines which flits have the right to go through the crossbar. The SA stage consists of a local switch arbiter (LSA) and a global switch arbiter (GSA). The LSA arbitrates among VCs and selects one VC for each input port. The GSA then determines which winning VC has the right to access the crossbar in this cycle. The flit from the winning VC is then read out and traverses the crossbar in the switch traversal (ST) stage. Finally, the flit travels to the downstream router in the link traversal (LT) stage.

III. VARIABLE PIPELINE STAGE ROUTER (VPSR)

Next, we present VPSR, a new router design which enables significant runtime power reduction while maintaining performance in the context of FinFET-based interconnection networks. VPSR adjusts the latency of router components by varying their back-gate bias voltages. The basic idea behind VPSR is to dynamically change the number of pipeline stages from the input ports to the output ports based on incoming traffic. Flits from more accessed ports, which are usually sources of contention, traverse fewer pipeline stages. Flits from less accessed ports, where most router resources are relatively free currently, have to pass through more pipeline stages. In other words, different flits may traverse varying-length pipeline stages when passing through the router. This has two advantages. Firstly, router performance is maintained because VPSR adapts its throughput to network traffic requirement at runtime. Secondly, significant leakage power can be saved by reverse-biasing the FinFET back-gates in infrequently accessed components. This scheme is only possible for FinFETs.

A. Critical Path Analysis and Optimization

To optimize both latency and power, we first analyze the critical path from the input port to the output port in the VPSR. Fig. 3 shows the critical path traversed by a head flit. It consists of the VC allocator and the LSA, followed by a buffer read (BR) stage, the GSA, and the crossbar. Arbiters are usually small circuits involving several logic gates and D flip-flops (DFFs), thus do not contribute too much to the critical path delay. The buffers and the crossbar, however, involve driving long wires and complex circuits and, hence, are the main contributors to the critical path

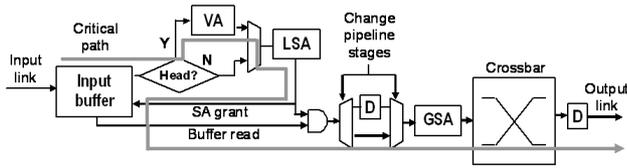


Fig. 3. VPSR critical path

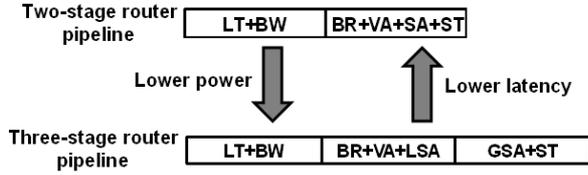


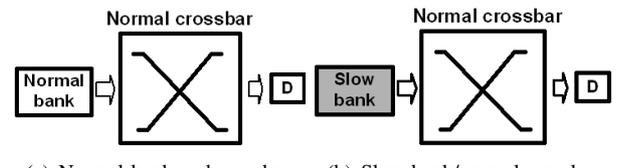
Fig. 4. VPSR pipeline stages

delay. This means that if the critical path is optimized under normally-biased LP-mode FinFETs, one more pipeline stage has to be added when LP-mode FinFETs in either the buffers or crossbar are reverse-biased. This is accomplished by inserting a flit-wide array of DFFs to separate the crossbar from the critical path. The increased slack accommodates the delay overhead incurred by reverse-biased LP-mode gates. For non-critical paths, gate sizes are optimized with reverse-biased LP-mode gates to further reduce their leakage power consumption. In this work, we optimize gate sizes along the critical path with normally-biased LP-mode FinFETs while meeting a target 1GHz clock frequency at the 32nm FinFET technology node.

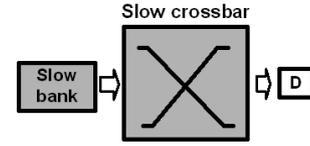
B. Variable Pipeline Stages

Based on the above observations, we propose a router with variable two-to-three pipeline stages. To illustrate its latency impact, Fig. 4 shows the VPSR pipeline through which a flit passes. BW and LT are combined to form a single stage. When a port is heavily accessed, a flit can head directly from the BR to the LT stage in a single cycle, without spending extra cycles in the conventional five-stage pipeline. This corresponds to the two-stage case. However, in case when a port is rarely visited or the power exceeds the budget, a flit has to spend one more cycle in the DFF array before the GSA stage. This leads to three pipeline stages in the router datapath. Correspondingly, this results in more slack, which allows the input buffer or crossbar to be reverse-biased to reduce leakage power. For an input port, VPSR gives a higher priority to the flit stored in the DFF array to access the GSA over the flits in the input buffer, which helps reduce average flit latency and, hence, improves throughput. On the other hand, the GSA gives each input flit an equal chance to access the crossbar.

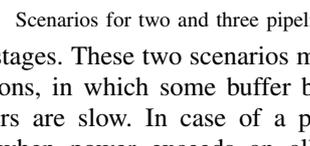
Fig. 5 illustrates the three scenarios where different back-gate bias voltages are applied to the input buffer and crossbar. Two modes of SRAM buffer banks are available in our design: normal bank and slow bank. A normal bank corresponds to normally-biased buffers ($V_{HI} = V_{DD}$ and $V_{LOW} = 0V$), whereas a slow bank corresponds to reverse-biased buffers ($V_{HI} > V_{DD}$ and $V_{LOW} < 0V$). In the first scenario, both the input buffer and crossbar are normally-biased, leading to the shortest critical path latency and, hence, two pipeline stages. In the second scenario, flits are read from a slow bank and pass through a normally-biased crossbar, leading to



(a) Normal bank and crossbar



(b) Slow bank/normal crossbar



(c) Slow bank and crossbar

Fig. 5. Scenarios for two and three pipeline stages

three pipeline stages. These two scenarios may coexist under normal conditions, in which some buffer banks are normal while the others are slow. In case of a power emergency (Scenario 3), when power exceeds an allowed threshold, VPSR turns itself to power saving mode by reverse-biasing all of the buffer banks and the crossbar. This minimizes overall leakage power consumption, but increases the number of pipeline stages to three for all flits passing through the router. A significant amount of dynamic power is also saved due to throughput degradation incurred by the longer pipeline. The scenario in which flits from a normal bank go through a reverse-biased crossbar is not allowed. This is to avoid extra leakage in the normal banks, because the reverse-biased crossbar results in three pipeline stages anyway.

IV. VPSR DYNAMIC POWER MANAGEMENT (VPSRDPM)

In this section, we present VPSRDPM, a framework used by VPSR to enable fine-grained DPM. Fine-grained control is made possible by partitioning the input buffers into several banks. Each bank can be switched between the normal and slow modes. The switching of bank modes is enabled by ABGB. To do this, each bank has its own voltage generator, which supplies required voltages to the back gates of FinFETs and is able to switch their voltage levels in a very short time. VPSRDPM uses a flow control mechanism to periodically update the ratio of the number of normal banks to the number of slow banks (normal-to-slow bank ratio) based on the incoming traffic condition. The flow control mechanism can proactively regulate the power consumption while meeting the traffic requirement. When the average power exceeds the power budget, VPSRDPM turns the router into the power emergency mode.

A. Voltage Generator

ABGB is of crucial importance for performing DPM in FinFET-based circuits because it is very effective and easy to implement. While popular techniques like dynamic voltage scaling (DVS) [16] can reduce active-mode power, they incur significant transition latency and energy overhead. ABGB, on the other hand, can switch the back-gate bias voltages of FinFETs very quickly, with little transition energy overhead.

In order to enable the ABGB technique for VPSRDPM, we incorporate voltage generators into the buffer banks and the crossbar. The CMOS voltage generator circuit from [17]

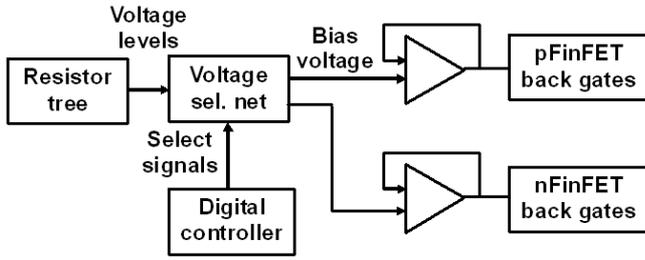


Fig. 6. Voltage generator block diagram

is adapted to FinFETs for this purpose. Fig. 6 shows the block diagram of the voltage generator. We optimize the design for fast transition (within 1ns or one clock cycle at 1GHz) from normally-biased ($V_{HI} = 1.0V$ and $V_{LOW} = 0V$) voltage levels to reverse-biased voltage levels ($V_{HI} = 1.2V$ and $V_{LOW} = -0.2V$), and vice versa. To accomplish this, the amplifier is scaled to tackle the load capacitance, which is equal to the sum of the interconnect capacitances and the back-gate capacitances of the FinFETs in the targeted circuit component. A range of voltage levels ($-0.2V \sim 1.2V$) can be generated and selected by digitally-controlled inputs. After the voltage transition, the voltage generator is turned off by disconnecting it from the power supply. Therefore, power is only dissipated during the voltage transition period.

The voltage generator consists of the following major blocks:

Resistor tree: This block consists of a series of resistors. These resistors divide the voltage range (V_H to V_L) and provide the required reference voltage levels. The resistance values are designed to be high so that the power consumed by the current flowing through it is minimized.

Voltage selection net: The function of the voltage selection net is to select a voltage level from the resistor tree and feed it to the unity-gain amplifier. Parallel pass transistors are used for this purpose, each connected to a tap in the resistor tree, providing various voltage levels for the unity-gain amplifier. Only one pass transistor is turned on at a time.

Unity-gain amplifier: The unity gain amplifier drives the back gates of FinFETs to the reference voltage level received from the voltage selection net. The sizes of transistors used in the amplifier are scaled appropriately to enable this. Two unity-gain amplifiers are required: one for p-type FinFETs and the other for n-type FinFETs.

Digital controller: This unit provides the digital selection signals for the voltage selection net. These signals are determined by the flow control mechanism.

Since energy is consumed every time the output of the voltage generator is switched, a corresponding amount of leakage energy should be saved in FinFETs to reach the breakeven point. For VPSR, the breakeven point is only four clock cycles. Our flow control mechanism updates the back-gate bias voltages only every 1,000 cycles. Thus, the energy saved by VPSR far outweighs the energy consumed by the voltage generator. The power consumption of the voltage generators is included in that of the router components in our simulations.

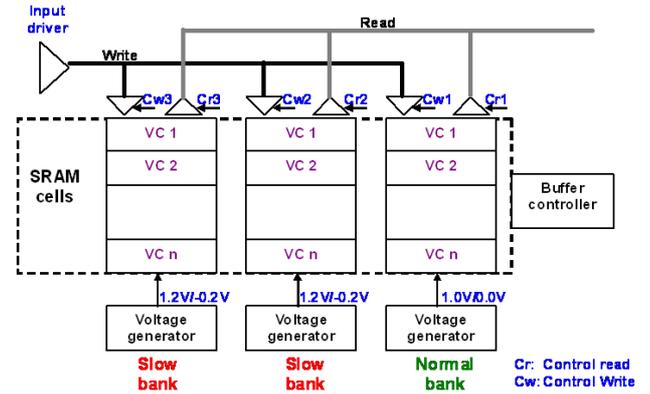


Fig. 7. Normal/slow bank structure

B. Normal Bank and Slow Bank Architecture

Fig. 7 shows the microarchitectural block diagram of the input buffer and buffer controller. As mentioned earlier, the input buffers are partitioned into normal banks and slow banks for fine-grained DPM. The motivation behind this is that memory cells are usually significant consumers of leakage power. In conventional use of a large pool of memory cells, each read or write operation only involves one row of memory cells, while the other rows are idle and dissipate leakage power. Thus, we partition the entire pool of memory into several banks with different bias voltages. The normal banks are biased at $V_{HI} = 1.0V$ and $V_{LOW} = 0V$, while the slow banks are biased at $V_{HI} = 1.2V$ and $V_{LOW} = -0.2V$. Although the banked-buffer scheme requires some extra wires and bitline decoders, the leakage power saved from the slow banks and the dynamic power saved from the shortened bitlines still outweigh the power consumed by them. Tri-state gates, which are controlled by the buffer controller, are used for controlling bank write and read operations. As shown in Fig. 7, each bank is shared among all VCs. The VC flit buffers are spread over the banks. The flits belonging to the same VC are written into the banks in a FIFO fashion. At low load, when most of the memory banks are free, VPSR sets most of its banks to slow banks. To avoid throughput degradation, normal banks are given higher priority to slow banks. Thus, if both normal banks and slow banks are available for an incoming flit, it will be written into a normal bank. When a flit is read from the input buffer, the buffer controller notifies VPSR which type of bank the next flit is coming from. Thus, VPSR is able to change its pipeline stages based on this information. A dual-port SRAM structure is used as a conventional input buffer to enable simultaneous write and read accesses of the bank. Thus, when a flit is read, the incoming flit is allowed to write into another memory address in parallel.

Fig. 8(a) shows the addressing mechanism used by the buffer controller. The VCs in the first bank are addressed by a lookup table, while the VCs in the other banks are simply sequentially addressed. Fig. 8(b) shows why the lookup table scheme is used for the first bank. Suppose a flit from VC1 stored in the first bank is being read. In parallel, another flit belonging to VC1 has arrived. If a sequential addressing scheme is used, the incoming flit cannot be

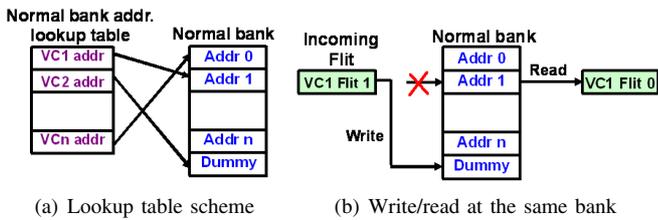


Fig. 8. Addressing mechanism employed in the first bank

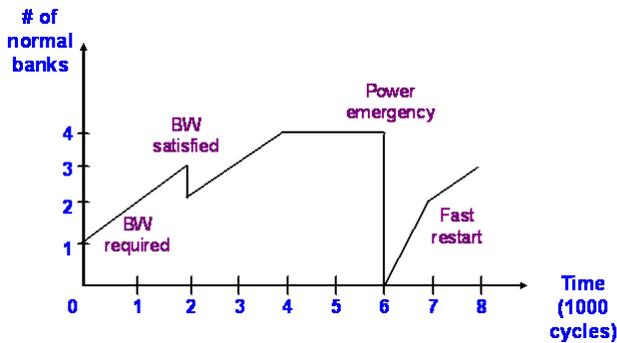


Fig. 9. VPSR flow control

written into this bank, because simultaneous write and read at the same memory location is not allowed. Therefore, this flit is required to be written to the next bank, which may be a slow bank and, hence, lead to a degradation in flit latency. As a solution to this problem, we add an extra dummy array besides the normal bank. This array has the same width as a flit, and can be in the form of an extra row of SRAM cells or a DFF array. Therefore, if a flit is being read from VC1, the incoming flit can be written into the dummy array. The lookup table then remaps the VC1 address to the dummy array and marks the original VC1 address as a free slot.

C. Flow Control Mechanism

The aim of the flow control mechanism in VPSRDPM is to determine the best normal-to-slow bank ratio. A good flow control would efficiently use bank resources to achieve high performance, but leave unused banks in the slow mode. The traffic patterns in CMPs do not always result in a high utilization at all input ports. This means that for some accessed ports, just a fraction of the banks need to be in normal mode to accommodate the incoming traffic. The other banks can be changed to slow mode to save leakage power. Our flow control mechanism dynamically adjusts the normal-to-slow bank ratio every 1,000 cycles to adapt to the traffic requirements at runtime. Thus, most flits spend the smallest possible time in the router pipeline. This allows the router's throughput to approach the all-normal-bank case, yet save significant amount of leakage power.

Fig. 9 shows an example of how the flow control mechanism works in VPSRDPM. Each unit on the horizontal axis is the update period (1,000 cycles). In the beginning, there is only one normal bank in each input buffer, the others being slow banks. Every 1,000 cycles, the flow control mechanism estimates the buffer utilization and power consumption [7] over the previous 1,000 cycles, and adjusts the normal-to-slow bank ratio. The four scenarios that the flow control mechanism has to handle at runtime are discussed next.

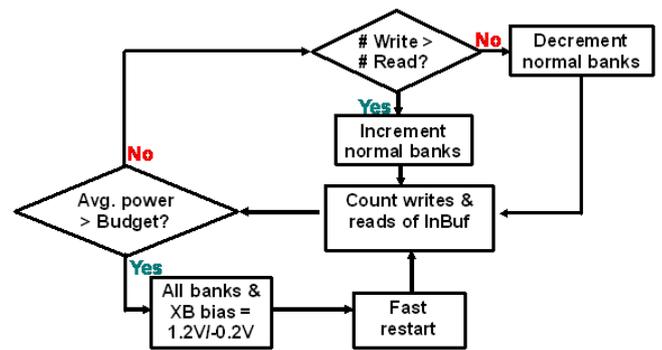


Fig. 10. Flow control for power management

BW required: This scenario occurs when the incoming traffic is faster than the outgoing traffic, which results in flits getting stuck in the input buffer. The existing normal-to-slow bank ratio is too small to handle the traffic load. Hence, one slow bank is changed to a normal bank to enable the increased throughput to clear out the flits in the slow bank.

BW satisfied: This scenario occurs when the number of outgoing flits is equal to or more than the number of incoming flits. Thus, the normal-to-slow bank ratio is more than required. To avoid excessive leakage power, the flow control mechanism decrements the number of normal banks by one, thus reducing throughput. However, to maintain the performance under light traffic, VPSR keeps at least one normal bank at each input buffer, if there is no power emergency.

Power emergency: The power emergency scenario occurs when the total router power exceeds the power budget. This budget is usually set by system cooling requirements, and is regulated by the operating system. The purpose of this budget is to restrict the system power from exceeding the limit that the system can support. Our flow control mechanism responds to this scenario by reverse-biasing the back gates of the FinFETs in all the memory banks as well as the crossbar switch. Hence, the leakage power and dynamic power are both greatly reduced.

Fast restart: In order to restore the router throughput quickly after a power emergency, VPSR has to readjust the normal-to-slow bank ratio to meet the incoming traffic pattern. To do this, the flow control mechanism remembers the normal-to-slow bank ratio used right before the power emergency period. Each time, when restoring from the power emergency mode, VPSR sets its normal-to-slow bank ratio to half the ratio right before the power emergency. This jump-to-half scheme has two advantages. Firstly, VPSR does not have to restart with all slow banks, thus its throughput can be restored quickly. Secondly, if the incoming traffic pattern changes, VPSR can re-adapt itself to the new traffic pattern instead of directly jumping to the all-normal-bank mode.

Fig. 10 shows the flowchart of the VPSR flow control mechanism. We have two counters for each input buffer: one counts the incoming flits and the other counts the flits read from the input buffer. On-line router power estimation [7] is used to estimate the power consumption in the previous cycle. Every update period, which is equal to 1,000 cycles, the number of buffer writes is compared with the buffer

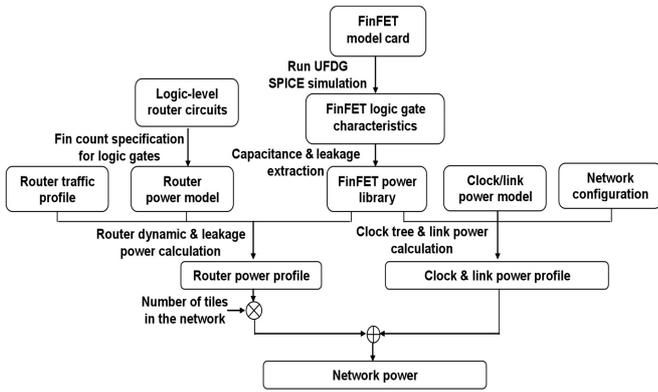


Fig. 11. ORION-FinFET power simulation flow

reads at each input port. If there are more flits written into the buffer than read out, the number of normal banks is incremented, otherwise it is decremented. In parallel, the average power is calculated for the past ten update periods. We average the power over the past ten update periods to avoid an instantaneous power overshoot. If the average power is greater than the power budget, VPSR enters the power emergency mode, and performs a fast restart after another update period.

V. GARNET-FINFET

GARNET [20] is used to model the power and performance of VPSR. GARNET is a cycle-accurate performance simulator for interconnection networks, and incorporates the original ORION router power model [21]. Hence, it provides a complete simulation framework to simulate the power consumption based on the traffic applied to the network. We have developed GARNET-FinFET, which is based on the original GARNET. GARNET-FinFET is able to simulate the power consumption and performance of a FinFET-based interconnection network. Moreover, GARNET-FinFET supports the VPSR structure, allowing the use of normal and slow banks, and enables its flow control mechanism.

A. ORION-FinFET and FinFET Power Library

ORION-FinFET [15] is revised from the original ORION [21], and is now incorporated in GARNET-FinFET. The most important feature of ORION-FinFET is that it supports the FinFET power libraries developed by us and several new power models for router components. The FinFET power library specifies the capacitance and leakage current values of logic gates for various FinFET operation modes and at different temperatures. Moreover, the library targets various FinFET technology nodes (45nm, 32nm, 25nm). Fig. 11 shows the simulation flow for ORION-FinFET. The simulator core comprises five major parts: router traffic profile, router power model, FinFET power library, clock/link power model, and network configuration. The first three generate the router power profile, while the last two and the FinFET power library generate the clock/link power profile. The network power consumption is obtained by combining the power consumption of routers with that of the clock tree and links.

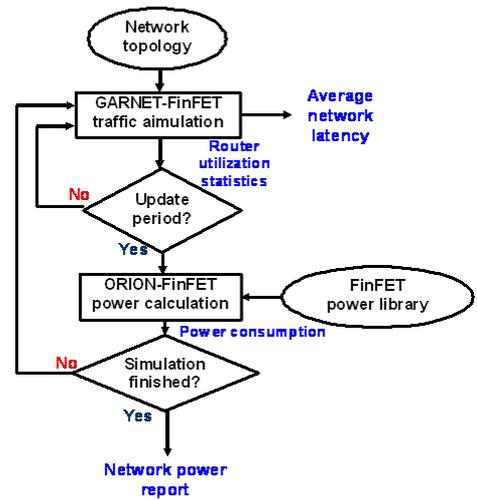


Fig. 12. GARNET-FinFET simulation flow

B. Simulation Flow

Fig. 12 illustrates how GARNET-FinFET works. The network topology is first specified by the user. GARNET-FinFET then injects flits into the network based on the given topology. When the update period is reached, GARNET-FinFET generates router utilization statistics of the previous period and calls ORION-FinFET to do power simulation. ORION-FinFET then uses the statistics to simulate the power consumption based on the FinFET power library. After power simulation is done, GARNET-FinFET checks if the overall simulation is completed. If so, GARNET-FinFET reports the network power. Otherwise, it returns to traffic simulation for the next update period and goes through the loop again. GARNET-FinFET calculates the average network latency as well, thereby allowing us to compare the performance for different traffic patterns and router configurations.

VI. EXPERIMENTAL RESULTS

In this section, we present experimental results of the proposed VPSR design. The network latency under different injection rates is explored first. Then, the comparison of network power consumption for different traffic patterns is presented. Finally, the power breakdown and latency of the network components are explored for four different cases.

A. Simulation Setup

Table I summarizes the key transistor-level design parameters we used for FinFETs. The values for L_G , T_{OX} and R_{SD} for the 32nm technology node are obtained from [11]. N_{BODY} and N_{DS} are typical values suggested by UFDG [22], which is an accurate process/physics-based double gate MOSFET model. T_{SI} is set to 7.0nm, which is within the typical range of $0.5L_G \sim 0.7L_G$.

Table II shows the network parameters along with the routing algorithm used. These numbers are configured into GARNET-FinFET. Each packet consists of five flits. The distance between two routers is assumed to be 1mm. If not specifically stated, the operating temperature is assumed to be 105°C, which is the peak temperature of the processors plus network of MIT's RAW chip [23]. The number of router ports ranges from three to five because the routers in the

TABLE I
32NM FINFET DESIGN PARAMETERS

FinFET parameters	Value
L_G (nm)	13
H_{FIN} (nm)	30
T_{Si} (nm)	7.0
T_{OX} (nm)	1.0
Channel doping, N_{BODY} (cm^{-3})	10^{15}
V_{DD} (V)	1.0
Parasitic series S/D resistance, R_{SD} ($\Omega\text{-}\mu m$)	170
Source/drain doping, N_{DS} (cm^{-3})	10^{20}

TABLE II
NETWORK PARAMETERS

Network parameters	Value
Frequency	1GHz
Technology	32nm FinFET
Topology	4x4 Mesh
Router ports	3~5
Flit size	128 bits
Number of VCs	12
Buffer size per VC	4
Link latency	1 cycle
Routing algorithm	Dimension-ordered X-Y

corner and along the sides have fewer input ports than the routers in the middle of the mesh. The number of simulation cycles is 100,000 and includes 1,000 cycles for warm-up. Synthetic traffic patterns are used in our simulations.

B. Network Latency vs. Injection Rate

Fig. 13 plots network latencies for VPSR and VPSR with power budget as a function of packet injection rate, comparing them with the all-normal-bank and all-slow-bank cases. The power budget of each router is set to the average power of the VPSR case, thus the excessive power consumed in VPSR is further reduced. The traffic pattern applied is uniform random, in which packets are sent to destinations in a uniform random fashion. It can be seen that the network latency of VPSR is very close to the all-normal-bank case. When the injection rate is 0.1 packets/node/cycle, the network latency of VPSR with power budget is only slightly increased. The all-slow-bank case quickly saturates as the injection rate increases.

C. Power Comparison for Different Traffic Patterns

Fig. 14(a) compares the power consumption of the entire network for the all-normal-bank, VPSR, VPSR with power budget, and all-slow-bank cases for three types of traffic patterns. Fig. 14(b) shows the power reduction obtained. These traffic patterns include uniform random, tornado, and bit-complement traffic [20], and were generated by GARNET-FinFET. The uniform random traffic was described earlier. In tornado traffic, packets are passed from router to router in a tornado-like manner. The bit-complement traffic sends packets from nodes to their bit-complement nodes. Different traffic patterns have different link and router port utilization rates, thus dissipate different amounts of power. The packet injection rate is fixed to 0.1/packets/node/cycle. It can be seen that on an average, VPSR alone is able to save 20% of power, as compared to the all-normal-bank case. Moreover, with a power budget, VPSR can reduce power by up to 29% for tornado traffic, and 25% on an average. Although the

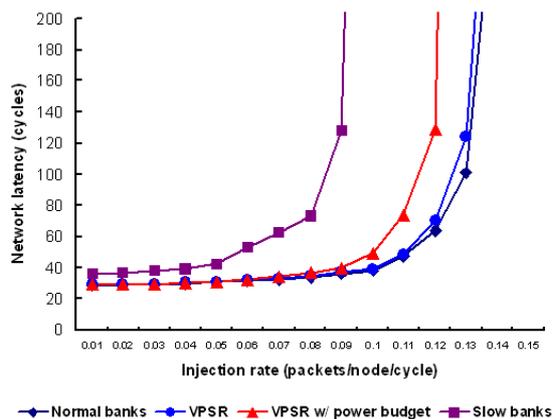
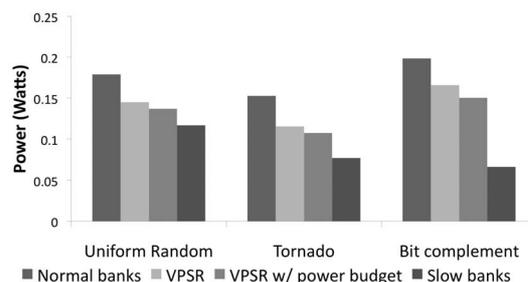
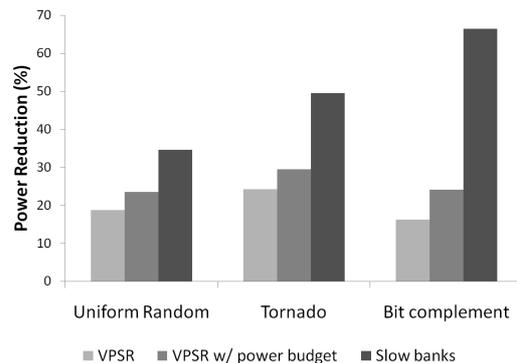


Fig. 13. Average network latency vs. injection rate



(a) Power comparison for different traffic patterns



(b) Power reduction ratio

Fig. 14. Power consumption vs. traffic

all-slow-bank case achieves the highest power reduction, its latency is very long. Therefore, this is only used when power emergency occurs.

D. Router Power Breakdown

Fig. 15 presents the power breakdown of the router under tornado traffic for the four cases. Fig. 16 plots the corresponding network delay. The power breakdown includes contributions from buffers, crossbar, arbiters, and local clock tree. Only the local clock tree within the router is included in this power breakdown, because the global clock distribution net should not be counted as part of the router power. As can be seen, VPSR reduces buffer power by 50%, with almost no degradation in latency. Under a power budget, VPSR further reduces crossbar power by 15%, thus reducing total power by 25%. The impact on latency is still not very significant.

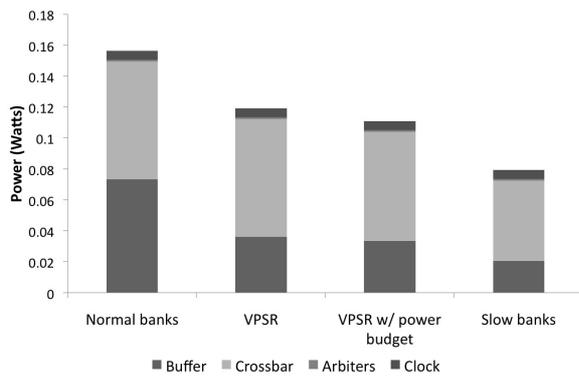


Fig. 15. Power breakdown for tornado traffic

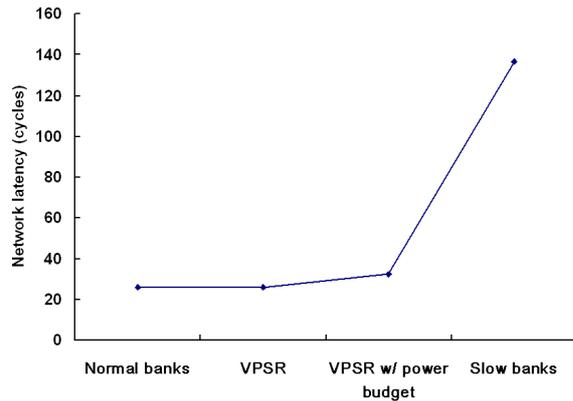


Fig. 16. Network latency for tornado traffic

VII. CONCLUSIONS

In this paper, we presented an in-depth design of a new router architecture, VPSR, suited to FinFET-based interconnection networks. VPSR targets power reduction while maintaining high network throughput. We proposed a flow control mechanism to regulate the normal-to-slow bank ratio for input buffer banks. We presented a voltage generator to enable fine-grained DPM through ABGB. We presented GARNET-FinFET which provides a complete platform for computer architects to quickly estimate the power of FinFET based interconnection networks at an early design stage. Experimental results show that VPSR is able to significantly reduce the power consumption by adapting to traffic, with little latency overhead.

VIII. ACKNOWLEDGMENTS

This work was supported by SRC under Contract No. 2007-HJ-1602 and 2008-HJ-1793, and NSF under Grant No. CNS-0613074.

REFERENCES

- [1] M. B. Taylor *et al.*, "Evaluation of the Raw microprocessor: An exposed-wire-delay architecture for ILP and streams," in *Proc. Int. Symp. Computer Architecture*, pp. 2-13, Jun. 2004.
- [2] K. Sankaralingam *et al.*, "Exploiting ILP, TLP, and DLP with the polymorphous TRIPS architecture," in *Proc. Int. Symp. Computer Architecture*, pp. 422-433, Jun. 2003.
- [3] K. Mai, T. Paaske, N. Jayasena, R. Ho, W. Dally, and M. Horowitz, "Smart memories: A modular reconfigurable architecture," in *Proc. Int. Symp. Computer Architecture*, pp. 161-171, Jun. 2000.
- [4] D. C. Pham *et al.*, "Overview of the architecture, circuit designs and physical implementation of the first generation cell processor," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 179-196, Jan. 2006.

- [5] Y. Hoskote *et al.*, "A 5-GHz mesh interconnect for a teraflops processor," *IEEE Micro*, vol. 27, no. 5, pp. 51-61, Sep.-Oct. 2007.
- [6] J. S. Kim, M. B. Taylor, J. Miller, and D. Wentzlaff, "Energy characterization of a tiled architecture processor with on-chip networks," in *Proc. Int. Symp. Low Power Electronics and Design*, pp. 424-427, Aug. 2003.
- [7] L. Shang, L.-S. Peh, and N. K. Jha, "PowerHerd: A distributed scheme for dynamically satisfying peak-power constraints in interconnection networks," *IEEE Trans. Computer-Aided Design*, vol. 25, no. 1 pp. 92-110, Jan. 2006.
- [8] L. Shang, L.-S. Peh, and N. K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *Proc. Int. Symp. High-Performance Computer Architecture*, pp. 91-102, Feb. 2003.
- [9] J. Kao, S. Narendra, and A. Chandrakasan, "Subthreshold leakage modeling and reduction techniques," in *Proc. Int. Conf. Computer-Aided Design*, pp. 141-148, Nov. 2002.
- [10] A. Muttreja, P. Mishra, and N. K. Jha, "Threshold voltage control through multiple supply voltages for power-efficient FinFET interconnects," in *Proc. Int. Conf. VLSI Design*, pp. 220-227, Jan. 2008.
- [11] "International Technology Roadmap for Semiconductors," <http://www.itrs.net>.
- [12] R. V. Joshi, K. Kim, R. Q. Williams, E. J. Nowak, and C.-T. Chuang, "A high-performance, low leakage, and stable SRAM row-based back-gate biasing scheme in FinFET technology," in *Proc. Int. Conf. VLSI Design*, pp. 665-672, Jan. 2007.
- [13] B. Swahn and S. Hassoun, "Gate sizing: FinFET vs. 32nm bulk MOSFETs," in *Proc. Design Automation Conf.*, pp. 528-531, Jul. 2006.
- [14] A. Muttreja, N. Agarwal, and N. K. Jha, "CMOS logic design with independent-gate FinFETs," in *Proc. Int. Conf. Computer Design*, pp. 560-567, Oct. 2007.
- [15] C.-Y. Lee and N. K. Jha, "FinFET-based power simulator for interconnection networks," *submitted to ACM J. Emerging Technologies in Computing Systems*.
- [16] N. K. Jha, "Low power system scheduling and synthesis," in *Proc. Int. Conf. Computer-Aided Design*, pp. 259-263, Nov. 2001.
- [17] J. Tschanz *et al.*, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1396-1402, Nov. 2002.
- [18] J. K. Ousterhout, G. T. Hamachi, R. N. Mayo, W. S. Scott, and G. S. Taylor, "MAGIC: A VLSI layout system," in *Proc. Design Automation Conf.*, pp. 152-159, Jun. 1984.
- [19] T.-J. King, "FinFETs for nanoscale CMOS digital integrated circuits," in *Proc. Int. Conf. Computer-Aided Design*, pp. 207-210, Nov. 2005.
- [20] N. Agarwal, T. Krishna, L.-S. Peh, and N. K. Jha, "GARNET: A detailed interconnection network model inside a full-system simulator," in *Proc. Int. Symp. Performance Analysis of Systems and Software*, Apr. 2009. Available for download from: <http://www.cs.wisc.edu/gems/download.html>.
- [21] H. Wang, X. Zhu, L.-S. Peh, and S. Malik, "ORION: A power-performance simulator for interconnection networks," in *Proc. Int. Symp. Microarchitecture*, pp. 294-305, Nov. 2002.
- [22] J. G. Fossum *et al.*, "A process/physics-based compact model for nonclassical CMOS device and circuit design," *Solid State Electronics*, vol. 48, no. 6, pp. 919-926, Jun. 2004.
- [23] L. Shang, L.-S. Peh, A. Kumar, and N. K. Jha, "Temperature-aware on-chip networks," *IEEE Micro*, vol. 26, no. 1, pp. 130-139, Jan.-Feb. 2006.