

# A Power Model for Routers: Modeling Alpha 21364 and InfiniBand Routers

Hang-Sheng Wang

Li-Shiuan Peh

Sharad Malik

{hangshen,peh,sharad}@ee.princeton.edu

Department of Electrical Engineering,

Princeton University,

Princeton, NJ08544.

## Abstract

*As interconnection networks proliferate to many new applications, a low-latency high-throughput fabric is no longer sufficient. Applications are becoming power-constrained. In this paper, we propose an architectural-level power model for interconnection network routers that will allow researchers and designers to easily factor in power when exploring architectural trade-offs. We applied our model to two commercial routers – the integrated Alpha 21364 router and the IBM 8-port 12X InfiniBand router, and show that the different micro-architectures lead to vastly different power consumption and distribution estimates.*

## 1 Introduction

Interconnection networks, historically used to connect processors and memories in large multiprocessors, are becoming prevalent in many new applications. They have been deployed to connect clusters of workstations [2], line cards of terabit Internet routers [4], server blades [10], and proposed as the fabric for on-chip networks [5, 14]. In many of these applications, a low-latency high-throughput fabric is no longer sufficient. Designers have to work within a tight power budget, and need to make architectural-level decisions considering power, in addition to performance.

Now, routers and links of an interconnection network are already taking up a significant portion of the overall system power. The integrated router and links of the Alpha 21364 microprocessor [11] consume about 20% of the total system power (25W out of total chip power of 125W), while the interconnection network circuitry on a router linecard takes up about 33% of the power consumed by the entire linecard [4]. Designers of an InfiniBand-enabled server blade allocated roughly the same power budget to the router and the microprocessor [10]. With the increasing demand for network bandwidth, power consumed by an interconnection network will be even more substantial.

While the power consumption of processing and memory elements are pretty well understood [3, 20], that of network elements have been largely neglected. This motivated us to develop an architectural-level power model for interconnection network routers, so researchers and designers can factor power estimates in easily when exploring different architectural trade-offs. This power model is part of our effort in providing a complete platform where designers and researchers can “pick, plug and play” different components to form myriad network architectures, both for chip-to-chip and on-chip networks. With our tool, users can run diverse communication workloads on different network architectures, and explore network power and performance rapidly in a single simulation environment.

In this paper, we propose a detailed router power model that is instantiated with architectural parameters. Section 2 first discusses the related work in power modeling of networks. Section 3 then explains our power model in detail. In Section 4, we apply our model to two commercial interconnection network routers that have distinct micro-architectures, the integrated Alpha 21364 router [11] and IBM 8-port 12X InfiniBand router [7] and show the vastly different power consumption characteristics. Section 5 concludes the paper, outlining our future directions.

## 2 Related work

Architectural-level power estimation is suitable for design space exploration because it is much faster than low-level power estimation tools such as PowerMill [16], while still giving reasonable accuracy. Several architectural-level power models have been proposed for microprocessors [3, 20], enabling a rich body of work on architectural-level power optimization. In interconnection networks, however, power models are scarce.

Patel *et al.* first proposed a power model for interconnection networks [12]. However, the model is not sufficiently detailed, with power estimates derived based on transistor count. In addition, the model is not instantiated with

architectural-level parameters, and thus cannot be used to explore trade-offs in router micro-architecture design.

Power models and simulators are available for other types of communication fabrics – there have been models proposed for other switch fabrics of Internet routers [17, 19], crossbars [6], buses [22] and on-chip bufferless switch-box networks [21]. The components of some of these fabrics are also present in interconnection networks (e.g. crossbars), and we leverage these prior components’ power models in our modeling of interconnection network routers.

Power-efficient interconnection networks is an emerging field. Prior research on power optimization of interconnection networks had to characterize router power using gate-level power estimation applied to detailed Verilog code [15]. It is our motivation that an architectural-level router power model will play a part in enabling research in power-efficient interconnection networks.

### 3 Power model

Many routers can be constructed from the basic building blocks of memory arrays, crossbars and arbiters. These components take up about 90% of the Alpha 21364 router area [1]. We thus develop router power models by deriving parameterized energy equations of these building blocks and tying them together.

Dynamic power is the primary source of power consumed in CMOS routers, which is formulated as  $P = E f_{clk}$ , and energy  $E = \frac{1}{2} \alpha C V_{dd}^2$ , with  $f_{clk}$  the clock frequency,  $\alpha$  the switching activity,  $C$  the switch capacitance, and  $V_{dd}$  the supply voltage. Power modeling consists of estimating switch capacitance  $C$  and switching activity  $\alpha$ .

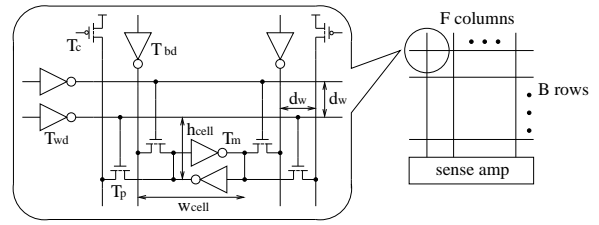
#### 3.1 Estimating capacitance

Table 1 lists the capacitance notations we use throughout our modeling.

**Table 1. Capacitance notations**

$C_g(T)$	gate capacitance of transistor/gate $T$
$C_d(T)$	diffusion capacitance of transistor/gate $T$
$C_a(T)$	$C_g(T) + C_d(T)$
$C_w(L)$	capacitance of metal wire of length $L$
$C_{in\_cnt}$	input node cap. of a crossbar connector
$C_{out\_cnt}$	output node cap. of a crossbar connector
$C_{ctr\_cnt}$	control node cap. of a crossbar connector
$C_{FF}$	switch capacitance of a flip flop
$C_{FC}$	clock capacitance of a flip flop

**FIFO buffers.** Buffers are typically implemented as SRAM arrays. Several architectural-level SRAM array



**Figure 1. A FIFO buffer with 1 read port and 1 write port.**

power models have been proposed [8, 23]. We adapt these models with some changes specific to router microarchitecture. For instance, a buffer with a dedicated port to the switch does not require tri-state output drivers. Figure 1 sketches the structure of a FIFO buffer, and Table 2 lists the model parameters and equations.

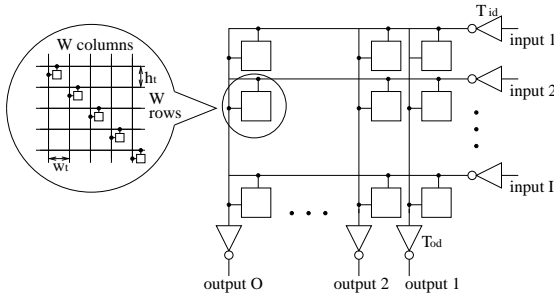
**Table 2. Model of FIFO buffers**

Architectural parameters	
$B$	buffer size in flits*
$F$	flit size in bits
$P_r$	number of buffer read ports
$P_w$	number of buffer write ports
Technological parameters	
$h_{cell}$	memory cell height
$w_{cell}$	memory cell width
$d_w$	wire spacing
Model equations	
wordline length	$L_{wl} = F(w_{cell} + 2(P_r + P_w)d_w)$
bitline length	$L_{bl} = B(h_{cell} + (P_r + P_w)d_w)$
wordline cap.	$C_{wl} = 2FC_g(T_p) + C_a(T_{wd}) + C_w(L_{wl})^*$
read bitline cap.	$C_{br} = BC_d(T_p) + C_d(T_c) + C_w(L_{bl})$
write bitline cap.	$C_{bw} = BC_d(T_p) + C_a(T_{bd}) + C_w(L_{bl})$
precharge cap.	$C_{chg} = C_g(T_c)$
memory cell cap.	$C_{cell} = 2(P_r + P_w)C_d(T_p) + 2C_a(T_m)$
sense amp energy	$E_{amp}$ from empirical model [23]

(\*) A flit is short for flow control unit, and is the smallest unit of buffer and channel allocation.

(\*)  $T_p$  is the pass transistor connecting bitlines and memory cells,  $T_{wd}$  is the wordline driver,  $T_{bd}$  is the write bitline driver,  $T_c$  is the read bitline precharge transistor,  $T_m$  is the memory cell inverter.

**Crossbar switch.** We consider two common crossbar implementations – multiplexer tree crossbar and matrix crossbar. In this paper, we explain just the matrix crossbar model. Figure 2 sketches the design.



**Figure 2. A matrix crossbar with  $I$  input ports and  $O$  output ports. The small square box represents a connector, which can be either a tri-state buffer or a transmission gate.**

In a matrix crossbar, input signals of a port propagate to the input nodes of the connectors in one row, the open/close states of those connectors determine which output lines get the input values. There are three components which may switch during data transfer: input lines, output lines, and connector control lines. Table 3 lists our model equations and parameters.

**Table 3. Model of matrix crossbar**

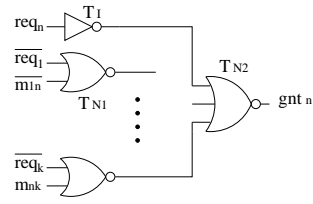
Architectural parameters	
$I$	number of crossbar input ports
$O$	number of crossbar output ports
$W$	port width in bits
Technological parameters	
$h_t$	track height
$w_t$	track width
Model equations	
input line length	$L_{in} = O \cdot W \cdot w_t$
output line length	$L_{out} = I \cdot W \cdot h_t$
input line cap.	$C_{xb\_in} = OC_{in\_cnt} + C_a(T_{id}) + C_w(L_{in})^*$
output line cap.	$C_{xb\_out} = IC_{out\_cnt} + C_a(T_{od}) + C_w(L_{out})$
control line cap.	$C_{xb\_ctr} = WC_{ctr\_cnt} + C_w(\frac{L_{in}}{2})^*$

(\*)  $T_{id}$  is the input driver,  $T_{od}$  is the output driver.

(\*) We use average length and assume control lines are along the same direction as input lines.

**Arbiters.** We model three kinds of arbiters: matrix arbiter, round-robin arbiter and queuing arbiter. In this paper, we explain just the matrix arbiter model.

A description of matrix arbiter can be found in [13]. For an arbiter with  $R$  requesters, its priorities can be represented by an  $R \times R$  matrix storing if a requester has higher priority than another. Only  $\frac{R(R-1)}{2}$  matrix elements need to be



**Figure 3. Grant generation logic**

stored in flip flops. Let  $req_i$  be the  $i$ th request,  $gnt_n$  be the  $n$ th grant,  $m_{ij}$  be the  $i$ th row,  $j$ th column element in the matrix, Equation (1) gives  $gnt_n$ , and Figure 3 shows its combinational logic.

$$gnt_n = req_n \cdot \prod_{i < n} (\overline{req_i} + \overline{m_{in}}) \cdot \prod_{i > n} (\overline{req_i} + m_{ni}) \quad (1)$$

From Equation (1) and Figure 3, we can derive the capacitance equations of request signals, grant signals, priority signals, internal nodes between the two levels of NOR gates, and the clock signal driving all priority flip flops, as shown in Table 4.

**Table 4. Model of matrix arbiter**

Architectural parameters	
$R$	number of requesters
Model equations	
request cap.	$C_{req} = C_a(T_I) + (R - 1)C_g(T_{N1}) + C_g(T_{N2})^*$
grant cap.	$C_{gnt} = C_d(T_{N2})$
priority cap.	$C_{pri} = C_{FF} + 2C_g(T_{N1})$
internal cap.	$C_{int} = C_d(T_{N1}) + C_g(T_{N2})$
clock cap.	$C_{clk} = C_{FC}$

(\*)  $T_{N1}$  is the first level NOR gate,  $T_{N2}$  is the second level NOR gate,  $T_I$  is the inverter.

Capacitance equations are used to compute energy consumption per switch of any component. For example,  $E_{wl} = C_{wl}V_{dd}^2$ . We omit  $\frac{1}{2}$  here because we count rising and falling as one switch for wordlines. In the following sections, we will use  $E_x$  notations without explanation.

### 3.2 Estimating maximum power

Maximum power  $P_{max}$  is the power consumed with maximum achievable switching activity at a certain network load. To simplify the illustration, we assume a simple wormhole router as shown in Figure 4. It has 5 input/output ports, with each input port having buffer space for 4 32-bit flits ( $B = 4, F = 32, P_r = 1, P_w = 1$ ); a  $5 \times 5$  crossbar of 32-bit width ( $I = 5, O = 5, W = 32$ ); and a 4:1 arbiter at each output port assuming flits do not u-turn ( $R = 4$ ).

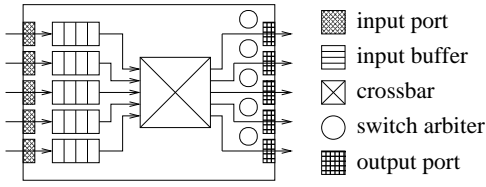


Figure 4. A wormhole router

Let  $E_{buffer}$ ,  $E_{crossbar}$ ,  $E_{arbiter}$  be the energy consumed by each input buffer, crossbar and arbiter in one cycle under  $P_{max}$ .

$$P_{max} = f_{clk}(5E_{buffer} + E_{crossbar} + 5E_{arbiter})$$

We assume network traffic is uniformly distributed across all router ports and define flit arriving rate  $P_f$  as the probability that each input port receives a flit in every cycle. We also assume input traffic equals output traffic, so in one cycle,  $P_f$  flits are written into each input buffer and  $P_f$  flits are read out from each input buffer. Maximum switching activity is achieved when all 32 bitlines and memory cells switch during read/write.

$$\begin{aligned} E_{buffer} &= P_f(E_{write} + E_{read}) \\ E_{write} &= E_{wl} + 32(E_{bw} + E_{cell}) \\ E_{read} &= E_{wl} + 32(E_{br} + 2E_{chg} + E_{amp}) \end{aligned}$$

$5P_f$  flits are sent to the crossbar, assuming the router uses deterministic routing, maximum switching activity is achieved when all  $5P_f$  flits go to different ports, i.e. there is no contention, and for each port, all 32 lines switch.

$$E_{crossbar} = 5P_f \cdot 32(E_{xb\_in} + E_{xb\_out})$$

The arbiter energy consists of that consumed due to the arbitration action, and that consumed by the clocking of flip flops. Let  $L$  be the average packet length in flits, because arbitration is only done for header flits and there is no contention, each arbiter receives one request in every  $\frac{L}{P_f}$  cycles. Maximum switching activity is achieved when all relevant priorities and all internal nodes switch, which is a loose upper bound. Clocking energy is summed over all flip flops and is a constant.

$$\begin{aligned} E_{arbiter} &= \frac{P_f}{L} E_{arbitration} + E_{clock} \\ E_{arbitration} &= (4-1)E_{pri} + 4(4-1)E_{int} + \\ &E_{req} + (E_{gnt} + E_{xb\_ctr}) \\ E_{clock} &= \frac{1}{2} \cdot 4(4-1)E_{clk} \end{aligned}$$

We add  $E_{xb\_ctr}$  to  $E_{gnt}$  because grant signals load crossbar control signals.

Since our power model is activity sensitive, it is also capable of computing average power, or more realistic power estimation based on real data trace. We have plugged our

power models into a dynamic power simulator for networks, providing a platform for exploring the impact of different microarchitectures and workloads on network power and performance.

## 4 Case studies

We estimate the average-case and worst-cast power of two commercial routers: the integrated Alpha 21364 router [11] and the IBM InfiniBand 8-port 12X router [7], and compare our results against power estimates provided by their designers. In our experiments, we compute average and maximum power at various  $P_f$ . Worst-case power is maximum power when  $P_f=1$ .

### 4.1 Router architectures

The two routers both have buffering at input ports. The Alpha 21364 router uses local and global arbiters, with local arbiters selecting a flit from multiple virtual channels at each input port, before forwarding requests to the global arbiters. As the Alpha 21364 router uses LRS (least recently served) policy, we assume matrix arbiters. We have no information on IBM InfiniBand router's arbitration, so we assume the same scheme as Alpha 21364. Later, we see this assumption does not affect overall estimation as arbiters consume a small fraction of total power. Table 5 lists their architectural and process parameters.

Table 5. Parameters of Alpha 21364 router and IBM InfiniBand router

	Alpha 21364	IBM InfiniBand
feature size	0.18 $\mu$ m	0.11 $\mu$ m
voltage	1.65V	1.2V
frequency	1.2GHz	250MHz
input ports	8	8
output ports	7	8
input buffer	$B=319/250/127/190^*$ $F=32, P_r=2, P_w=1$	$B=256, F=128,$ $P_r=1, P_w=1$
crossbar	$I=8, O=5, W=32$	N/A
central buffer	N/A	$B=2560, F=512,$ $P_r=2, P_w=2$
local arbiter	$R=19$	$R=4$
global arbiter	$R=7$	$R=7$

(\* different buffer size of inter-processor ports, cache ports, memory controller ports and I/O ports, respectively.

Technological parameters such as  $h_{cell}$ ,  $d_w$  and transistor

sizes are measured from floorplan, if available, scaled from Wattach [3], or computed using Cacti [18].

The key difference between the two routers lies in their switch fabric architecture. The Alpha 21364 router has two  $8 \times 5$  crossbars as its switch fabric, instead of one  $8 \times 7$  crossbar. Each crossbar covers a subset of output ports and two read ports of the input buffers connect to distinct crossbars.

On the other hand, the IBM InfiniBand router uses a central buffer rather than a crossbar as the switch fabric. It is implemented as a pipelined shared memory [9] with 4 banks, each 1 flit wide, and total buffer size is 160K bytes. As our buffer model is sufficiently generic, it can be slightly augmented to model pipelined shared memory.

Although a pipelined shared memory is more sophisticated than a traditional SRAM array, in terms of total energy, a port consecutively reading/writing 4 banks is equivalent to it simultaneously reading/writing all 4 banks. This is equivalent to a read/write to an unbanked SRAM array with the same size and organization, plus some extra energy consumed by the additional components of a pipelined shared memory. Those additional components include the pipeline registers between memory banks and the input/output crossbars to/from the shared memory. We already have crossbar models and flip flop models, so we incorporate these in our modeling of a pipelined shared memory. This approximation is not valid for cycle-by-cycle power simulation, but it is appropriate for maximum power estimation.

## 4.2 Results and analysis

We applied our model for estimating the average and maximum power consumed by the two routers. Maximum power is estimated by deriving maximum possible switching activity factors given a flit arrival rate, while average power assumes 50% switching probability. Figure 5 shows maximum and average power estimated by our model for both routers, along with a breakdown of the maximum power consumed by the various router components, at varying flit arrival rate  $P_f$ . Our results provide several insights into router power consumption.

**Which component dominates?** For the IBM InfiniBand router, switch fabric (central buffer) power is dominant, which is more than 90% of total power (see Figure 5(e)). For the Alpha 21364 router, input buffers contribute 46-61% of total power, switch fabric (crossbar) consumes less (26-35%), but still significant power (Figure 5(f)). In both cases, the largest memory is also the largest power consumer. This rule has been shown true for most general-purpose microprocessors. Our results confirm that it holds true too for interconnection network routers.

**Which component is not important?** Arbiter power is negligible for the IBM InfiniBand router. It is not as negligible for Alpha 21364 router, and even shows some signifi-

cance at low  $P_f$ . This difference is caused by the different router microarchitectures.  $E_{arbiter}$  increases quadratically with  $R$ , as the Alpha 21364 router's local arbiter has  $R=19$ , its arbiter power is comparable with crossbar power at low  $P_f$  (28% of total power). We also test other types of arbiters and draw the same conclusion.

**Sensitivity to network load.** For both routers, buffer power and crossbar power are approximately linear to  $P_f$ , which is intuitive, because both buffers and crossbars are data path components, and data path power is approximately linear to the amount of data passing by. For the IBM InfiniBand router, the power breakdown is insensitive to  $P_f$ , since both input buffers and switch fabric are SRAM arrays, and are similarly sensitive to  $P_f$ . As expected, the Alpha 21364 router does not have this characteristic.

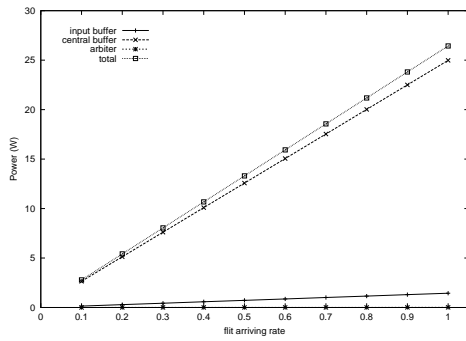
**Comparison with designers' estimates.** We obtained preliminary power estimates from the designers of these two routers. The Alpha 21364 designers estimate the Alpha router core to consume a maximum of  $7.6W^1$ . The power number is estimated by applying average power density of previous-generation chips to the chip area occupied by the router, along with a certain share of the global power. The designers estimate this to be within 10% of actual maximum power. As for the IBM InfiniBand 8-port 12X switch, the designers based their power estimates on circuit-level capacitance estimates, together with estimates of average activity factors based on previous-generation routers. They estimate the router core dissipating 11W with links consuming 20W. These are average power estimates.

Our estimated worst-case power of the Alpha 21364 router (100% flit arrival in Figure 5(c)) is 5.36W. This is within ballpark of the designers' estimate. As for the IBM InfiniBand switch, the 11W average power estimate from the designers corresponds to our average power estimate at 60% flit arrival rate (see Figure 5(b)). These comparisons are clearly not rigorous. We hope to be able to validate our model in detail with industry data and low-level power estimation tools, and are currently in the process of collaborating with router designers towards this goal.

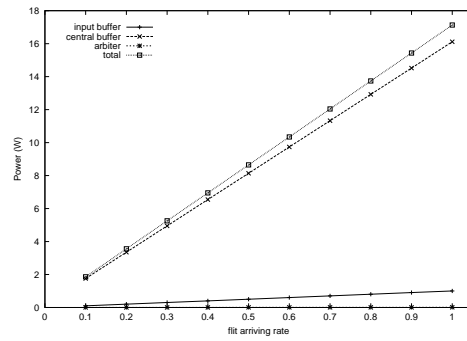
**What's missing?** Several components of the router core are not yet modeled, such as routing logic, routing table, and the InfiniBand interface to microprocessors. We also ignore link power in this work since we are only modeling routers. Link power is especially significant in chip-to-chip networks (consuming about 60% of total network (router and link) power in both the InfiniBand and Alpha routers)<sup>2</sup>. We will be developing parameterized power mod-

<sup>1</sup>Alpha 21364 designers estimate the integrated router and links to dissipate 25W out of the total chip power of 125W, with the router core consuming 7.6W, link circuitry (drivers, pad logic) consuming 13.3W, clocks taking up 2W and miscellaneous circuitry the remaining.

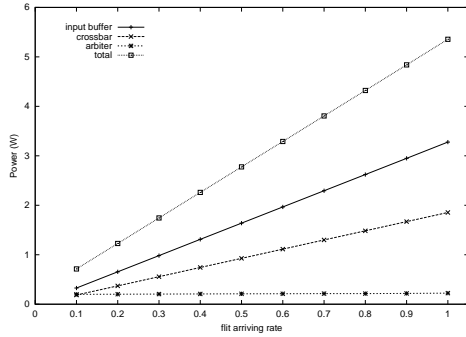
<sup>2</sup>In on-chip networks, we found routers to be the predominant power consumer (85% on average) in our experiments instead of links. The difference is largely because most chip-to-chip links incur approximately the



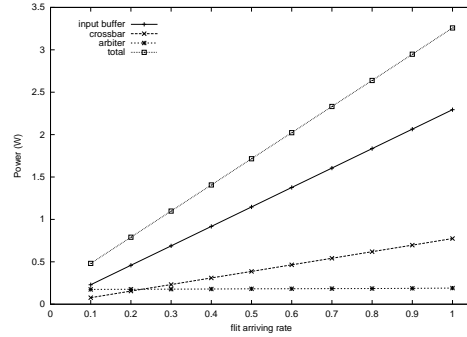
(a) maximum power of IBM InfiniBand router



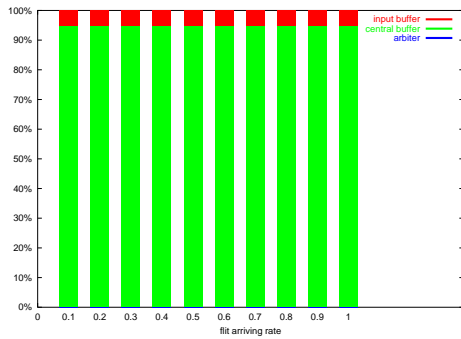
(b) average power of IBM InfiniBand router



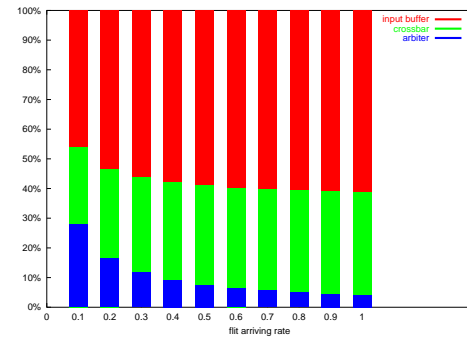
(c) maximum power of Alpha 21364 router



(d) average power of Alpha 21364 router



(e) maximum power breakdown of IBM InfiniBand router



(f) maximum power breakdown of Alpha 21364 router

**Figure 5. Router power estimation results**

els for links as well in the future. Another avenue of future research is the estimation of static power in addition to dynamic power.

---

same amount of power regardless of actual link utilization, while on-chip wire power is highly traffic-sensitive.

## 5 Conclusions and future work

We present our architectural-level power model for interconnection network routers, which can be used to model a large range of router microarchitectures. We use our model to estimate the average and maximum power of two on-chip

routers: the integrated Alpha 21364 router and IBM InfiniBand 8-port 12X switch. Preliminary results match power estimates given by router designers in the magnitude level. Our experiments show that buffers are the largest power hog in routers; crossbar switch, if present, consumes less but significant power; and power consumed by arbiters is negligible under high network load. Our results also show that router microarchitecture has a huge impact on the power distribution of a router.

In the future, we will validate our models rigorously and plug them into a dynamic network simulator to build a re-targetable and flexible network simulation environment that will enable network architects to perform power and performance trade-offs in a single framework. We will also extend our power model to support links and additional router components as well as static power.

## Acknowledgments

The authors are very grateful to Craig Stunkel of IBM T.J. Watson, for providing architectural details and power numbers of the InfiniBand 8-port 12X switch. We also wish to thank Peter Bannon, Allen Baum and Aaron Spink of the Alpha 21364 design team for patiently answering our numerous questions on the router.

## References

- [1] P. Bannon. Alpha 21364 (EV7). [http://www.eecs.umich.edu/vlsi\\_seminar/f01/slides/bannon.pdf](http://www.eecs.umich.edu/vlsi_seminar/f01/slides/bannon.pdf).
- [2] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W.-K. Su. Myrinet – a gigabit-per-second local-area network. *IEEE Micro*, 15(1), 1995.
- [3] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proc. International Symposium on Computer Architecture*, 2000.
- [4] W. Dally, P. Carvey, and L. Dennison. The Avici terabit switch/router. In *Proc. Hot Interconnects 6*, 1998.
- [5] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proc. Design Automation Conference*, 2001.
- [6] G. Essakimuthu, N. Vijaykrishnan, and M. J. Irwin. An analytical power estimation model for crossbar interconnects. Technical Report CSE-02-009, Penn State University, Department of Computer Science and Engineering, 2002.
- [7] IBM. IBM InfiniBand 8-port 12x switch. <http://www-3.ibm.com/chips/products/infiniband/>.
- [8] M. B. Kamble and K. Ghose. Analytical energy dissipation models for low-power caches. In *Proc. International Symposium on Low Power Electronics and Design*, 1997.
- [9] M. Katevenis, P. Vatsolaki, and A. Efthymiou. Pipelined memory shared buffer for VLSI switches. In *Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 1995.
- [10] Mellanox Technologies Inc. Mellanox performance, price, power, volume metric (PPPV). <http://www.mellanox.com/products/shared/PPPV.pdf>.
- [11] S. S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb. The Alpha 21364 network architecture. *IEEE Micro*, 22(1), 2002.
- [12] C. S. Patel, S. M. Chai, S. Yalamanchili, and D. E. Schimmel. Power constrained design of multiprocessor interconnection networks. In *Proc. International Conference on Computer Design*, 1997.
- [13] L.-S. Peh and W. J. Dally. A delay model and speculative architecture for pipelined routers. In *Proc. International Symposium on High Performance Computer Architecture*, 2001.
- [14] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, and A. Sangiovanni-Vincentelli. Addressing the system-on-a-chip interconnect woes through communication-based design. In *Proc. Design Automation Conference*, 2001.
- [15] L. Shang, L.-S. Peh, and N. K. Jha. Power-efficient interconnection networks: Dynamic voltage scaling with links. *Computer Architecture Letters*, 1(2), 2002.
- [16] Synopsys Corporation. PowerMill datasheet. [http://www.synopsys.com/products/etg/powermill\\_ds.html](http://www.synopsys.com/products/etg/powermill_ds.html).
- [17] A. G. Wassal and M. A. Hasan. Low-power system-level design of VLSI packet switching fabrics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(6), 2001.
- [18] S. J. Wilton and N. P. Jouppi. An enhanced access and cycle time model for on-chip caches. Technical Report 93/5, DEC Western Research Laboratory, 1994.
- [19] T. T. Ye, L. Benini, and G. D. Micheli. Analysis of power consumption on switch fabrics in network routers. In *Proc. Design Automation Conference*, 2002.
- [20] W. Ye, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin. The design and use of SimplePower: A cycle-accurate energy estimation tool. In *Proc. Design Automation Conference*, 2000.
- [21] H. Zhang, M. Wan, V. George, and J. Rabaey. Interconnect architecture exploration for low-energy reconfigurable single-chip DSPs. In *Proc. IEEE Computer Society Workshop on VLSI*, 1999.
- [22] Y. Zhang, W. Ye, and M. J. Irwin. An alternative architecture for on-chip global interconnect: Segmented bus power modeling. In *Proc. Asilomar Conference on Signals, Systems, and Computers*, 1998.
- [23] V. Zyuban and P. Kogge. The energy complexity of register files. In *Proc. International Symposium on Low Power Electronics and Design*, 1998.