
A POWER MODEL FOR ROUTERS: MODELING ALPHA 21364 AND INFINIBAND ROUTERS

AS INTERCONNECTION NETWORKS PROLIFERATE TO MANY NEW APPLICATIONS, A LOW-LATENCY HIGH-THROUGHPUT FABRIC NO LONGER SUFFICES. AN ARCHITECTURAL-LEVEL POWER MODEL FOR INTERCONNECTION NETWORK ROUTERS WILL LET RESEARCHERS AND DESIGNERS EASILY FACTOR IN POWER WHEN EXPLORING ARCHITECTURAL TRADEOFFS.

..... Interconnection networks, historically used to connect processors and memories in large multiprocessors, are becoming prevalent in many new applications. Designers have deployed interconnection networks to connect workstation clusters,¹ terabit Internet router line cards,² and server blades.³ Recently, designers have used interconnection networks as the fabric for on-chip networks.⁴ In many applications, a low-latency, high-throughput fabric no longer suffices. Designers must work within a tight power budget and make architectural-level decisions that consider power in addition to performance.

Routers and links of an interconnection network already consume a significant portion of the overall system power. The integrated router and links of the Alpha 21364 microprocessor⁵ consume about 20 percent of the total system power (25 W out of total chip power of 125 W), while the interconnection network circuitry on a router line card accounts for about 33 percent of the line card's total power consumptions. Designers of an InfiniBand-enabled server blade allocated roughly the same power budget to the router

and the microprocessor.³ With increasing demand for network bandwidth, the power that an interconnection network consumes will be even more substantial.

Although researchers and designers have pretty well understood processing and memory elements' power consumption,⁶ they have largely neglected that of network elements. This motivated us to develop an architectural-level power model for interconnection network routers, so researchers and designers can factor in power estimates easily when exploring different architectural tradeoffs. This power model is part of our effort to provide a complete network simulation platform where designers and researchers can pick, plug, and play different components to form myriad network architectures, run diverse communication workloads, and rapidly explore network power and performance alternatives. See sidebar on Orion.

Related work

Architectural-level power estimation is suitable for design space exploration because it is much faster than low-level power estimation

Hang-Sheng Wang
Li-Shiuan Peh
Sharad Malik
Princeton University

tools such as PowerMill from Synopsys corporation, while still giving reasonable accuracy. Researchers have proposed several architectural-level power models for microprocessors,⁶ 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.⁷ However, the model is not sufficiently detailed; it derives power estimates based on transistor count. Also, the model is not instantiated with architectural-level parameters, and thus you cannot use it to explore tradeoffs in router microarchitecture design.

Power models and simulators are available for other types of communication fabrics.^{8,9} Some components of these fabrics (such as crossbars) are also present in interconnection networks, and we leverage these prior power models in our interconnection network router modeling.

Power-efficient interconnection networks is an emerging field. Prior research on the power optimization of interconnection networks had to characterize router power using gate-level power estimation applied to detailed Verilog code.¹⁰ An architectural-level power model for routers will play an important role in enabling research on power-efficient interconnection networks.

Power model

Designers can construct many routers from the following basic building blocks: memory arrays, crossbars, and arbiters. These components consume about 90 percent of the Alpha 21364 router area. We thus develop router power models of these building blocks by deriving parameterized energy equations tying them together.

For current process technologies, dynamic power is the primary power source consumed in CMOS circuits. We formulate power as $P = E f_{\text{clk}}$, and energy $E = 1/2 \cdot \alpha C V_{\text{dd}}^2$, with clock frequency f_{clk} , switching activity α , switch capacitance C , and supply voltage V_{dd} . Power modeling consists of estimating C and α .

Estimating capacitance

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

Table 1. Capacitance notations.

Notation	Description
$C_g(T)$	Gate capacitance of transistor/gate T
$C_d(T)$	Diffusion capacitance of transistor/gate T
$C_a(T)$	Sum of $C_g(T) + C_d(T)$
$C_w(L)$	Capacitance of metal wire of length L
$C_{\text{in_cnt}}$	Input node capacitance of a crossbar connector
$C_{\text{out_cnt}}$	Output node capacitance of a crossbar connector
$C_{\text{ctr_cnt}}$	Control node capacitance of a crossbar connector
C_{FF}	Switch capacitance of a flip-flop
C_{FC}	Clock capacitance of a flip-flop

Orion

Orion is a power-performance simulator for interconnection networks that allows architects to explore the effect of different traffic workloads on diverse network architectures, enabling rapid power-performance trade-offs at the architectural-level.¹ Orion is built atop the Liberty Simulation Environment, a general framework that constructs a simulator from a microarchitectural specification.² A key component of Orion is the architectural-level parameterized power models that we have presented in this article. Orion will be released in 2003, and its status can be followed on the Orion Web site at <http://www.ee.princeton.edu/~peh/orion.html>.

1. H.-S. Wang et al. "Orion: A Power-Performance Simulator for Interconnection Networks," *Proc. 35th Ann. Int'l Symp. Microarchitecture (Micro-35)*, IEEE CS Press, 2002.
2. M. Vachharajani et al. "Microarchitectural exploration with Liberty," *Proc. 35th Ann. Int'l Symp. Microarchitecture (Micro-35)*, IEEE CS Press, 2002.

FIFO buffers. Designers typically implement buffers as static RAM (SRAM) arrays. Some on-chip networks, such as the Raw microprocessor,⁴ use shift registers to form first-in, first-out buffers due to the less demanding buffer space requirement of static networks. We model both, but explain just the SRAM-array-based model here.

Researchers have proposed several architectural-level SRAM array power models.¹¹ We adapt these models and make them more fine-grained and also make changes specific to router microarchitecture. For instance, a FIFO buffer does not need a decoder. Figure 1 (next page) sketches the structure of a FIFO buffer, and Table 2 lists the model parameters and equations.

Crossbar switch. We consider two common crossbar implementations—multiplexer tree and matrix. Here, we explain just the matrix

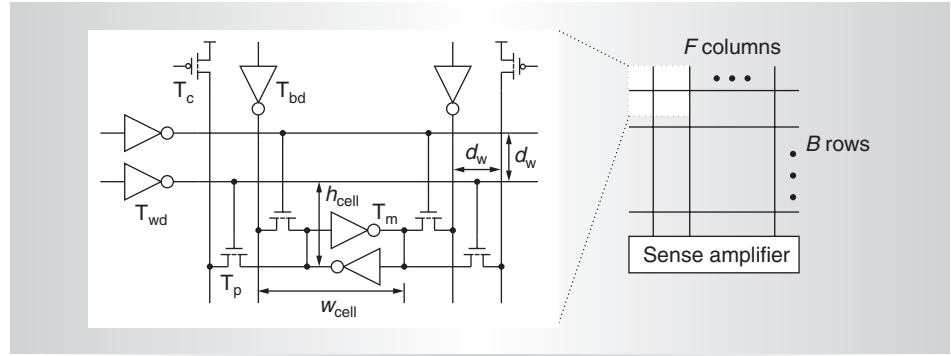


Figure 1. FIFO buffer with one read and one write port.

Table 2. FIFO buffer model.

Characteristic	Description
Architectural parameters	
B	Buffer size in flits*
F	Flit size in bits
P_r	No. of buffer read ports
P_w	No. of buffer write ports
Technology parameters	
h_{cell}	Memory cell height
w_{cell}	Memory cell width
d_w	Wire spacing
Model equations	
Word line length	$L_{wl} = F[w_{cell} + 2(P_r + P_w)d_w]$
Bit line length	$L_{bl} = B[h_{cell} + (P_r + P_w)d_w]$
Word line capacitance	$C_{wl} = 2FC_g(T_p) + C_a(T_{wd}) + C_w(L_{wl})^{**}$
Read bit line capacitance	$C_{br} = BC_d(T_p) + C_d(T_c) + C_w(L_{bl})$
Write bit line capacitance	$C_{bw} = BC_d(T_p) + C_a(T_{bd}) + C_w(L_{bl})$
Precharge capacitance	$C_{chg} = C_g(T_c)$
Memory cell capacitance	$C_{cell} = 2(P_r + P_w)C_d(T_p) + 2C_a(T_m)$
Sense amplifier energy	E_{amp} from empirical model***

*"Flit" is short for flow control unit; it is the smallest unit of buffer and channel allocation.

** T_p is the pass transistor connecting bit lines and memory cells; T_{wd} , the word line driver; T_{bd} , the write bit line driver; T_c , the read bit line precharge transistor; and T_m , the memory cell inverter.

*** V. Zyuban and P. Kogge, The Energy Complexity of Register Files, Proc. Int'l Symp. on Low Power Electronics and Design, 1998.

crossbar model; Figure 2 shows its design.

In a matrix crossbar, data from an input port propagate to the input ends of the connectors belonging to the same row; the open/close states of those connectors determine which output ports receive the input data. Three components can switch during data transfer: lines for input, output, and con-

nectors control. Table 3 lists the model parameters and equations.

Arbiters. We model three types of arbiters: matrix, round-robin, and queuing. Here, we explain just the matrix arbiter model.

For an arbiter with R requesters, you can represent its priorities by an $R \times R$ matrix, with a 1 in row i and column j if requester i has higher priority than another requester j , and 0 if otherwise. This method requires just $R(R-1)/2$ matrix elements in flip-flops. Let req_i be the i th request; gnt_m , the m th grant; and m_{ij} , the i th row and j th column element in the matrix. Using these variables,

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

Figure 3 shows its combinational logic.

From this equation and Figure 3, we can derive the capacitance equations of request, grant, and 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 (on p. 30).

All our power models are parameterized. Architectural and technology are the two parameter categories. In addition to those listed, technology parameters also include all involved transistor sizes. Detailed technology parameters are not always available for a given device; when these are unavailable, we use three approaches to obtain them:

- Measure the parameters from the floor plan or its photo, if available. We mea-

sured the Alpha 21364's crossbar size from its floor plan photo.

- You can determine the size of some transistors using their load capacitance and timing requirements.
- Use default values, many of which come from the Wattch framework.⁶

So a network designer can use our model before determining all the low-level details. But knowing these technology parameters can greatly improve the model's accuracy, as we will see in the following sections.

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

Estimating switching activity and maximum power

In this section, we use maximum power as an example to demonstrate how to estimate switching activity using a probabilistic approach.

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 (next page). It has five input/output ports, with each input port having buffer space for four 32-bit flits ($B = 4$; $F = 32$; $P_r = 1$; $P_w = 1$); a 32-bit wide, 5×5 crossbar ($I = 5$; $O = 5$; $W = 32$); and a 4:1 arbiter at each output port. assuming flits do not make u-turns ($R = 4$).

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

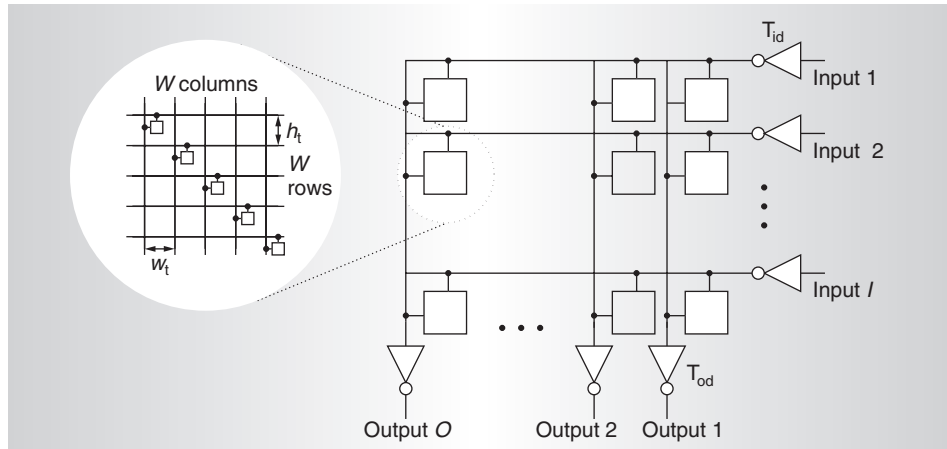


Figure 2. Matrix crossbar with I input ports and O output ports. The small square box represents a connector, which can be either a tristate buffer or a transmission gate.

Table 3. Matrix crossbar model.

Characteristic	Description
Architectural parameters	
I	No. of crossbar input ports
O	No. of crossbar output ports
W	Port width in bits
Technology parameters	
h_t	Track height
w_t	Track width
Model equations	
Input line length	$L_{in} = OW_{wt}$
Output line length	$L_{out} = IW h_t$
Input line capacitance	$C_{xb_in} = OC_{in_cnt} + C_a(T_{id}) + C_w(L_{in})^*$
Output line capacitance	$C_{xb_out} = IC_{out_cnt} + C_a(T_{od}) + C_w(L_{out})$
Control line capacitance	$C_{xb_ctr} = WC_{ctr_cnt} + C_w(L_{in}/2)^{**}$

* T_{id} is the input driver, T_{od} is the output driver.

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

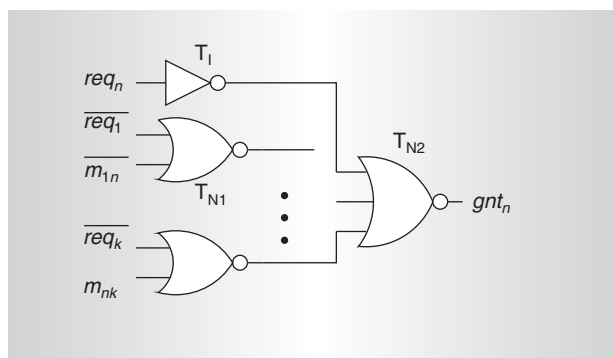


Figure 3. Grant generation logic.

Table 4. Matrix arbiter model.

Characteristic	Description
Architectural parameter	
R	No. of requesters
Model equations	
Request capacitance	$C_{\text{req}} = C_a(T_1) + (R - 1)C_g(T_{N1}) + C_g(T_{N2})^*$
Grant capacitance	$C_{\text{gnt}} = C_d(T_{N2})$
Priority capacitance	$C_{\text{pri}} = C_{\text{FF}} + 2C_g(T_{N1})$
Internal capacitance	$C_{\text{int}} = C_d(T_{N1}) + C_g(T_{N2})$
Clock capacitance	$C_{\text{clk}} = C_{\text{FC}}$

* T_{N1} is the first-level NOR gate; T_{N2} , the second-level NOR gate; and T_1 , the inverter.

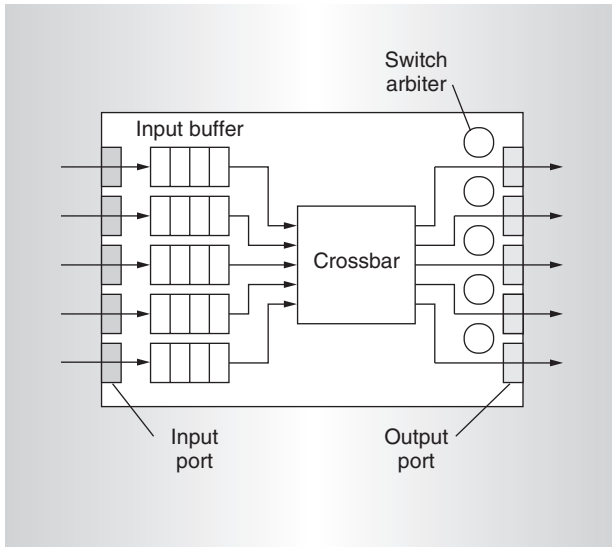


Figure 4. Wormhole router.

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

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. You achieve maximum switching activity when all 32 bit lines and memory cells switch during read/write.

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

Assume that $5P_f$ flits go to the crossbar in each cycle. You achieve maximum switching activity when all $5P_f$ flits go to different ports; for example, there is no contention, and for each input/output port, all 32 lines switch.

$$E_{\text{crossbar}} = 5P_f \times 32(E_{\text{xb_in}} + E_{\text{xb_out}})$$

The arbiter energy consists of the energy consumed by the arbitration action, and the energy that the clocking of flip-flops consumes. Let L be the average packet length in flits, because there is no contention and arbitration is only necessary for header flits; so each arbiter receives one request in every L/P_f cycle. A loose upper bound of maximum switching activity comes from the situation when all relevant priorities and all internal nodes switch, although this is not achievable. Clocking energy is summed over all flip-flops and is a constant.

$$\begin{aligned} E_{\text{arbiter}} &= P_f / L E_{\text{arbitration}} + E_{\text{clock}} \\ E_{\text{arbitration}} &= (4 - 1)E_{\text{pri}} + 4(4 - 1)E_{\text{int}} + E_{\text{req}} + (E_{\text{gnt}} \\ &\quad + E_{\text{xb_ctr}}) \\ E_{\text{clock}} &= 1/2 \times 4(4 - 1)E_{\text{clk}} \end{aligned}$$

We add $E_{\text{xb_ctr}}$ to E_{gnt} because grant signals load crossbar control signals. You can similarly derive the switching activity for average power.

Case studies

We validated our model against the Raw microprocessor, as the “Model Validation” sidebar describes. We then apply our model to two commercial routers: the integrated Alpha 21364 router⁵ and the IBM InfiniBand 8-port 12X router.¹² We estimate their average and worst-case power, comparing our results against power estimates provided by their designers. In our experiments, we compute average and maximum power at various flit arrival rates P_f . Worst-case power is maximum power when $P_f = 1$.

Router architectures

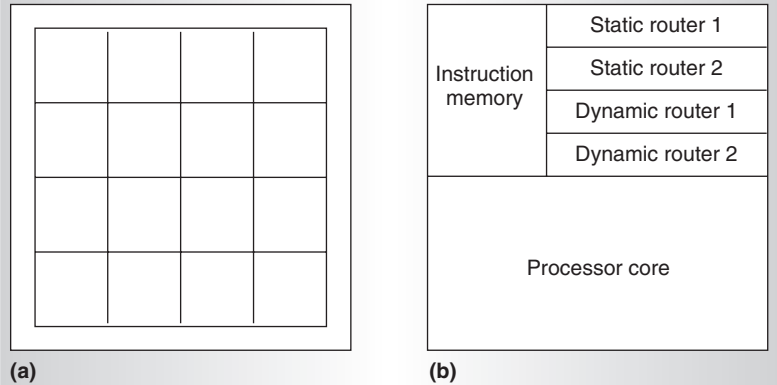
The two routers both have buffers at input ports. The Alpha 21364 router uses a local/global arbitration scheme. The local arbiter associated with an input port selects a flit from multiple virtual channels, then sends a request to the global arbiter associated with the output port. As the Alpha 21364 router

Model validation

We have validated our power models against the MIT Raw microprocessor¹, whose block diagram is shown in the Raw chip figure. The MIT Raw microprocessor is a computation fabric with 16 tiles on a chip. There is one processor core in each tile, and four on-chip networks across the whole chip: general dynamic network, memory dynamic network and two static networks. These network resources connect the 16 tiles as four 2-D meshes. Despite their functionality differences, the four networks have similar router architectures. Each router has its input buffers, crossbars and control logic. The input buffer is also called Network Input Block (NIB) by the Raw group and is implemented as shift registers, rather than SRAMs as in most other networks. The crossbar has a mux-based implementation. Unlike dynamic networks, which employ dynamic arbitration to control the traffic, the two static networks have an 8K-64 instruction memory on each tile, which stores the pre-compiled scheduling of their resources.

By the time this article is written, we have successfully validated our crossbar model and shift register model against their crossbars and NIBs. Although the instruction memory is not as simple as a FIFO buffer, our FIFO buffer model is based on a more general SRAM power model and we validate that SRAM model against the instruction memory, too. As the control logic is tightly coupled with the processor pipeline logic in the Raw networks, we have not extracted the arbitration logic and the arbiter model remains unvalidated.

Table 5 shows our validation results, with relative error ranging from 3 percent to 26 percent. We only show the comparison of the theoretical maximum switch capacitance since it is trivial to get accurate power estimation once we can estimate capacitance accurately. The third column lists the values given by our power models, and the second column lists the values reported by the Raw group, which were obtained through



A Raw chip (a) with 16 tiles and a block diagram of one tile (b).

Table 5. Model validation against the MIT Raw microprocessor.

Component	capacitance – Raw (pF)	capacitance – model (pF)	Relative error
Crossbar	74	72	3%
NIB	77	74	4%
Instruction memory (read)	239	285	20%
Instruction memory (write)	216	273	26%

a circuit-level power estimation tool, the IBM databook or Chipbench, an IBM EDA placement tool. This validation study with Raw networks allowed us to better tune our estimation of technology parameters from architectural parameters, improving our models' accuracy. We will be validating our models' estimates with Raw's actual chip measurements in the near future.

Reference

1. M.B. Taylor et al., "The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs, *IEEE Micro*, vol. 22, no. 2, Mar-Apr. 2002, pp. 25-35.

uses a least recently served (LRS) policy, we assume matrix arbiters. We have no information on IBM InfiniBand router's arbitration, so we assume the same scheme as that of the Alpha 21364. We will see this assumption does not affect overall estimation as arbiters consume a small fraction of total power. Table 6 lists the routers' architectural parameters and technology parameters.

The key difference between the two routers lies in their switch fabric architecture. The Alpha 21364 router has two 8×5 crossbars

as its switch fabric, instead of one 8×7 crossbar. Each crossbar covers a subset of output ports and two read ports of each input buffer connection to distinct crossbars.

On the other hand, the InfiniBand router uses a central buffer rather than a crossbar as the switch fabric. IBM implemented the buffer as a pipelined shared memory with four banks, each one flit wide; the total buffer size is 160 Kbytes.

We slightly augment our generic buffer model to represent a pipelined shared mem-

Table 6. Parameters of Alpha 21364 router and IBM InfiniBand router.

Characteristic	Alpha 21364	IBM InfiniBand
Feature size (μm)	0.18	0.11
Voltage (V)	1.65	1.2
Frequency (MHz)	1,200	250
No. of input ports	8	8
No. of output ports	7	8
Input buffer	$B = 319/250/127/190$ flits* $F = 32, P_r = 2, P_w = 1$	$B = 256$ flits, $F = 128$, $P_r = 1, P_w = 1$
Crossbar	$I = 8, O = 5, W = 32$	NA
Central buffer	NA	$B = 2,560, F = 512$, $P_r = 2, P_w = 2$
Local arbiter	$R = 19$	$R = 4$
Global arbiter	$R = 7$	$R = 7$

*These numbers are the different buffer size of interprocessor, cache, memory controller, and I/O ports.

ory. A pipelined shared memory is much more complex than a conventional SRAM array in terms of timing. But a port consecutively reading or writing four banks consumes the same total energy as a port simultaneously reading or writing all four banks. This is equivalent to a read/write to an unbanked SRAM array with the same size and organization, plus the extra energy consumed by additional components of a pipelined shared memory, namely pipeline registers and input/output crossbars. We already have crossbar and flip-flop models, so we incorporate them 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 and average power estimation.

Results and analysis

We use the approach described earlier to estimate the two routers' average and maximum power. Figure 5 (next page) shows maximum and average power for both routers, along with the component breakdown of maximum power. 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 percent of total power, as Figure 5e

shows. For the Alpha 21364 router, input buffers contribute 46 percent to 61 percent of total power, switch fabric (crossbar) consumes less (26 percent to 35 percent), but still significant power, as Figure 5f shows. In both cases, the largest memory is also the largest power consumer. This rule has proven true for many general-purpose microprocessors. Our results confirm that it also holds 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 significance at low P_f . The router microarchitectures cause this difference. 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 percent of total power). We also tested other arbiter types and drew 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 their power is approximately linear to the amount of data passing by. For the IBM InfiniBand router, the power breakdown is insensitive to P_f because both input buffers and switch fabric are SRAM arrays and are identically 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 these two routers' designers. Alpha 21364 designers estimated the integrated router and links to dissipate 25 W out of the total chip power of 125 W, with the router core consuming 7.6 W; link circuitry (drivers and pad logic), 13.3 W; clocks, 2 W; and miscellaneous circuitry consuming the rest. These numbers are obtained by applying average power density of previous-generation chips to the chip area that the router occupies, along with a certain share of global power. The designers estimated their power estimates to be within 10 percent of actual maximum power. As for the IBM InfiniBand 8-port 12X switch, the designers based their power estimates on esti-

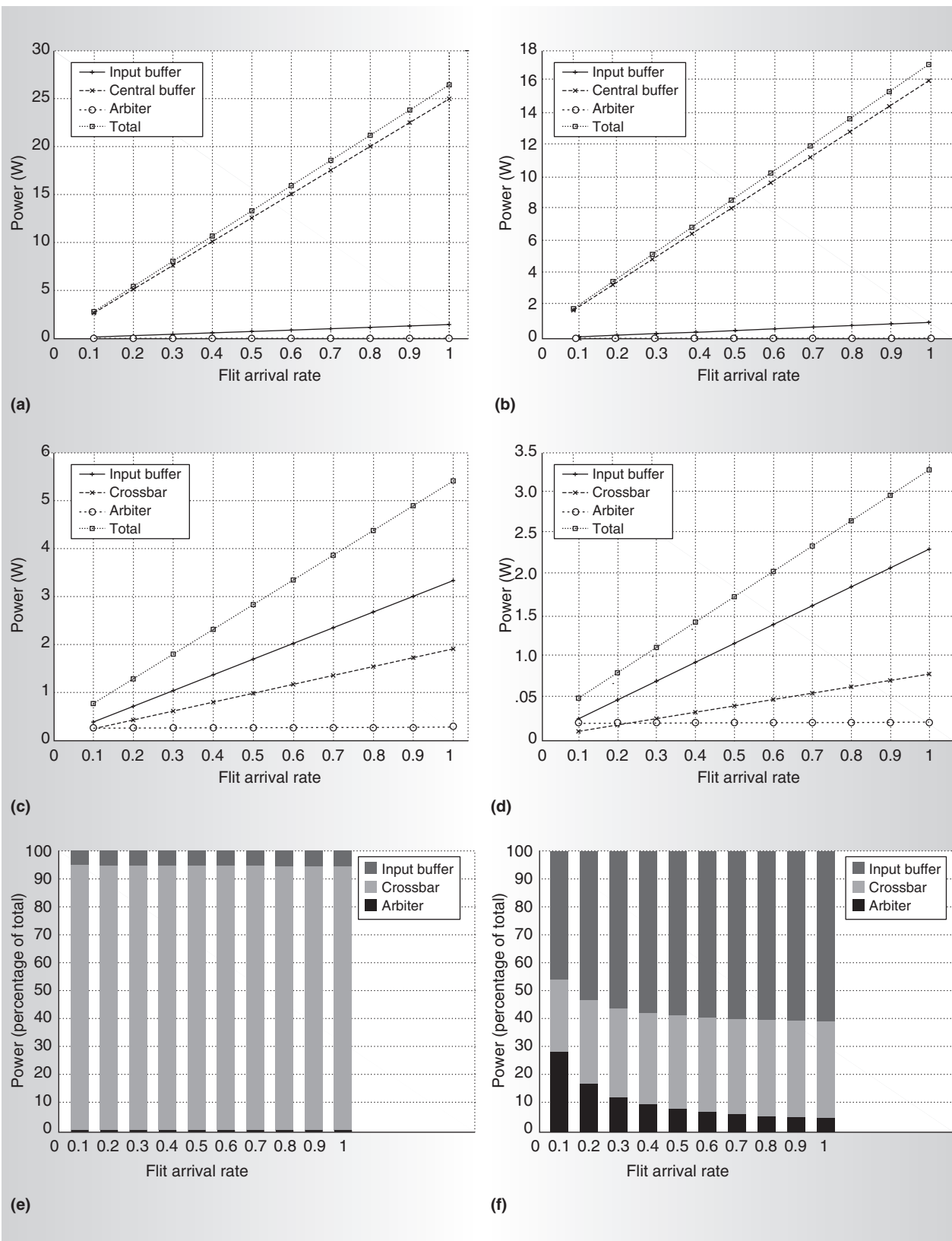


Figure 5. Router power estimation results.

mates of circuit-level capacitance and estimates of average activity factors culled from previous-generation routers. They estimated that on average the router core dissipated 11 W, and links consume 20W.

Our estimated worst-case power for the Alpha 21364 router (100-percent flit arrival in Figure 5c) is 5.36 W. This is within the ballpark of the designers' estimate. As for the IBM InfiniBand switch, the 11 W average power estimate from designers corresponds to our average power estimate at a 60 percent flit arrival rate, shown in Figure 5b). Because the conditions under which designers' estimates were obtained are unknown, this is just an approximate comparison.

What's missing? Several components of the router core, such as the clock tree, routing logic and routing table, are not yet modeled. We ignore link power in this work, focusing on just router power modeling. Link power is especially significant in chip-to-chip and board-to-board networks (consuming about 60 percent of total network power in both the IBM InfiniBand and the Alpha routers). In on-chip networks, our experiments identified routers as the dominant power consumer (85 percent on average) instead of links. The difference is largely because the more complex transmitter and receiver circuitry of chip-to-chip and board-to-board links incur significantly more power than repeated on-chip wire tracks. We also ignore leakage power in this work.

We present our architectural-level power model for interconnection network routers, that is applicable to a diverse range of router microarchitectures in on-chip, chip-to-chip and board-to-board networks. We validate our model against the MIT Raw microprocessor and use our model to estimate the average and maximum power of two commercial routers: the integrated Alpha 21364 router and the IBM InfiniBand 8-port 12X switch. Our experiments show that buffers are the largest power hog in routers; crossbar switch, if present, consumes less but significant power; and arbiter power is negligible under high network load. Our results also show that router microarchitecture has a huge impact on router power distribution. MICRO

Acknowledgments

We are grateful to Craig Stunkel of IBM T.J. Watson Research for providing architectural details and power numbers for the InfiniBand 8-port 12X switch. We thank Peter Bannon, Allen Baum, and Aaron Spink of the Alpha 21364 design team for patiently answering our numerous questions on the router. We greatly thank Jason Kim of the MIT Raw group who was instrumental in our validation effort, providing us with detailed architectural and technology parameters of the Raw networks.

References

1. N.J. Boden et al., "Myrinet: A Gigabit-per-Second Local Area Network," *IEEE Micro*, vol. 15, No. 1, Feb. 1995, pp. 29-36.
2. W. Dally, P. Carvey, and L. Dennison, "The Avici Terabit Switch/Router," <http://www.avici.com>
3. Mellanox Technologies Inc., "Mellanox Performance, Price, Power, Volume Metric (PPPV)," <http://www.mellanox.com/products/shared/PPPV.pdf>.
4. M.B. Taylor et al., "The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs," *IEEE Micro*, vol. 22, no. 2, Mar.-Apr. 2002, pp. 25-35.
5. S.S. Mukherjee et al., "The Alpha 21364 Network Architecture," *IEEE Micro*, vol. 22, No. 1, Jan.-Feb. 2002, pp. 26-35.
6. D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *Proc. Int'l Symp. Computer Architecture (ISCA 00)*, ACM Press, 2000, pp. 83-94.
7. C.S. Patel et al., "Power Constrained Design of Multiprocessor Interconnection Networks," *Proc. Int'l Conf. Computer Design (ICCD 97)*, IEEE CS Press, 1997, pp. 408-416.
8. A.G. Wassal and M.A. Hasan, "Low-Power System-Level Design of VLSI Packet Switching Fabrics," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no.6, June 2001, pp. 723-738.
9. G. Essakimuthu, N. Vijaykrishnan, and M.J. Irwin, *An Analytical Power Estimation Model for Crossbar Interconnects*, tech. report CSE-02-009, Dept. of Computer Science and Engineering, Penn State Univ., University Park, Pa., 2002.

10. L. Shang, L.-S. Peh, and N.K. Jha "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks," Proc. 9th Int'l Symp. High-Performance Computer Architecture (HPCA), 2003.
11. M.B. Kamble and K. Ghose, "Analytical Energy Dissipation Models for Low-Power Caches," Proc. Int'l Symp. Low Power Electronics and Design (ISPLED 97), ACM Press, 1997, pp. 143-148.
12. IBM. "IBM InfiniBand 8-port 12x Switch," <http://www-3.ibm.com/chips/products/infiniband/> (now protected)

Hang-Sheng Wang is a PhD student in the department of Electrical Engineering at Princeton University. His research interests include power modeling and simulation and power-aware interconnection networks. Wang has a BS and MS in Microelectronics from Tsinghua University, Beijing, China.

Li-Shiuan Peh is an assistant professor of electrical engineering at Princeton University since 2002, leading a research group exploring power-aware interconnection networks. Her research interests include interconnection networks, computer architecture and network-

ing in general. Peh has a BS in Computer Science and Information Systems from National University of Singapore, and a PhD in Computer Science from Stanford University. She is a member of the IEEE and has been awarded the 2003 NSF Career award.

Sharad Malik is a professor of the department of Electrical Engineering at Princeton University. His research interests include design tools for embedded computer systems, synthesis, and verification of digital systems. Malik has a BS in Electrical Engineering from Indian Institute of Technology, New Delhi, India, and an MS and PhD in Computer Science from the University of California, Berkeley. He has served on the program committees of DAC, ICCAD and ICCD. He served as the technical program co-chair for DAC in 2000 and 2001 and Panels Chair for 2002. He is on the editorial boards of the Journal of VLSI Signal Processing, Design Automation for Embedded Systems, and IEEE Design and Test. He is a fellow of the IEEE.

For further information on this or any other computing topic, visit our Digital Library at <http://computer.org/publications/dlib>.

Look for these topics in IEEE Computer Society magazines this year

Computer

Agile Software Development
Piracy & Privacy

IEEE Computer Graphics & Applications

3D Reconstruction & Visualization

Computing in Science & Engineering

The End of Moore's Law

IEEE Design & Test

Clockless VLSI Design

IEEE Intelligent Systems

AI & Elder Care

IEEE Internet Computing

The Semantic Web

IT Professional

Financial Market IT

IEEE Micro

Hot Chips 14

IEEE MultiMedia

Computational Media Aesthetics

IEEE Software

Software Geriatrics:
Planning the Whole Life Cycle

IEEE Security & Privacy

Digital Rights Management

IEEE Pervasive Computing

Smart Spaces



computer.org/publications