Low-Power System-Level Design of VLSI Packet Switching Fabrics

A.G. Wassal, *Member, IEEE,* and M.A. Hasan, *Senior Member, IEEE*

Abstract— **System-level design of packet switching fabrics focuses on performance metrics and rarely considers the physical requirements which are usually addressed later at the circuit-level. However, low-power dissipation has become a major requirement in such fabrics dictated by the requirements of emerging applications and by the recent advances in fabrication and VLSI technologies. This paper proposes a framework for system-level design of packet switching fabrics that integrates performance specifications along with physical requirements and constraints. Moreover, realistic traffic models are used to derive the transition activity and the packet arrival and departure events needed for power estimation. Physical requirements are defined by an architectural model for power dissipation based on the stochastic traffic model, models for silicon area, chip count and I/O pins which provide a complete system-level specification of the fabric. Performance constraints are also derived from the stochastic traffic model. This framework formulates and solves the power optimization problem subject to those physical and performance constraints as an integer non-linear optimization problem. The results obtained emphasize the importance of traffic-driven system-level optimization and show the efficiency of this framework as a system-level design space exploration tool.**

I. Introduction

Recent advances in VLSI technology such as scaling the clock frequency and higher memory bandwidth have made the design of packet switching fabrics with very high aggregate traffic rates possible. Very dense memories and logic circuits are being packed to provide for as many connections as possible and to achieve very low cell loss probabilities. However, such designs often produce very hot chips and low-power techniques are only employed at the circuit-level and attain little improvement. On the other hand, new applications are emerging with very scarce energy sources, special chip and system packaging requirements and complex cooling and heat dissipation processes. One such situation is evident in *global broadband networks* where the switches are used on-board low-earth-orbit satellites, LEOs, to move the network backbone into the sky and to provide integrated services to virtually everywhere on earth. Supplied from the energy stored by the satellite solar cells, on-board switches would have to dissipate as little power as possible and would require special packaging to prevent the ill effects of radiation and ions on the electronic circuits. This type of packaging would also make cooling and heat dissipation a very difficult task. Consequently, all of those factors have to be considered during early design stages and the design for low-power dissipation should be at the system-level where the largest possible improvement can be achieved.

Designing switching fabrics at the system-level has always been based on optimizing certain criteria related to performance metrics such as throughput, cell loss probability and mean cell delay [1, 2]. Only a handful of studies in the literature attempted to consider practical and physical requirements in switch design as well. Shaikh *et al.* compared the counts of the crosspoints and the pin-limited chips in two switching fabrics; namely the Shufflenet and Banyan networks [3]. In [4], Coppo *et al.* proposed a methodology for evaluation and optimization of Clos networks, used as ATM switching fabrics, based on component count and interconnection cost as physical cost requirements for a given connection blocking probability. Zegura presented a comparison of different switching fabrics based on the count of pin-limited chips needed by each fabric [5]. Shi *et al.* explored the optimization of the Knockout switch hardware cost given certain constraints on the quality of service [6]. Most recently, Schultz studied the performance limits of shared memory architectures using experimental results and technology trend curves [7]. Moreover, almost all of those studies used certain traffic profiles or a certain set of given quality of service parameters that would make the analysis of the physical requirements simpler and more tractable.

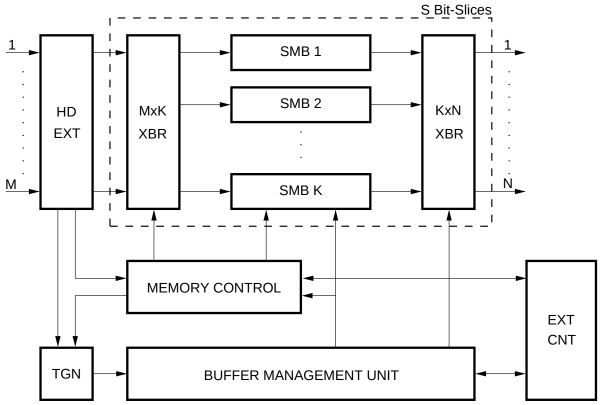
In this paper, a framework for system-level design of packet switching fabrics is proposed. This framework uses a traffic-based approach to determine the transition activity[1] for digital signals in a shared memory IP/ATM switching fabric and to formulate an expression for the power dissipation. The same traffic model is also used to determine the dimensions of the switch and consequently formulate the appropriate expressions for the required silicon area and access rates. Section II focuses on the shared memory architecture and its use in IP/ATM switching. An analytical traffic model for the fabric is also introduced in that section and used to formulate the performance constraints. The analytical model is used in section III to devise the power dissipation requirements in the shared memory architecture using real traffic statistics and the *General Sampled-Bit Regression* model, GSBR.

Section IV addresses the formulation of some other system-level physical and performance constraints such as expressions for the required silicon area and access rates.

Appeared in *IEEE Trans. Computer Aided Design*, pp. 723-738, June 2001.

A.G. Wassal was with the University of Waterloo and is now with PMC-Sierra, Burnaby, BC V5A 4V7, Canada. M.A. Hasan is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: {wassal, ahasan}@vlsi.uwaterloo.ca.

[1] Transition activity refers to the probability of signal transitions between logic 0 and 1 which results in dynamic power dissipation in the system.



HD EXT	:	Header Extraction and Conversion.
EXT CNT	:	External Control and Accounting.
TGN	:	Tag Generator.
$M \times K$ XBR	:	$M \times K$ Ingress Crossbar.
$K \times N$ XBR	:	$K \times N$ Egress Crossbar.
SMB	:	Shared Memory Block.

Fig. 1. A shared memory switch architecture.

In section V, the system-level design problem is viewed as an integer non-linear optimization problem with power as the objective function to be minimized and the other requirements as the bounding constraints. Numerical results are also presented and discussed in that section. Finally, some conclusions are given in section VI.

II. The Shared Memory Switch Architecture

In this section, we will consider a shared memory architecture for an $M \times N$ switch, where M and N are the numbers of input and output ports respectively. The link rates are all V cells per second. In this architecture, a physical memory pool is used by all logical queues, hence the term "shared". Logical queues are implemented using address linked lists and managed by a separate memory controller. Each linked list represents a logical queue associated with a certain output port. Given a specific cell loss probability, shared memory architectures have the advantage of a smaller buffer size per output port due to the sharing effect among ports. It can also be modeled as an output buffering architecture which guarantees the throughput if the hardware can support it. On the other hand, the main problem with the shared memory model is that the memory access rate has to be higher than or equal to the value $(M+N)V$. However, the architecture proposed here, as shown in figure 1, resolves that by dividing the shared memory into K blocks accessible from all inputs and outputs which allows each block to have an access rate of only N operations per cell time. This is because, on the worst case, outputs may have to access the same block simultaneously while the inputs can be controlled to write to different idle memory blocks. The memory access rate can now be independent of the number of input ports provided that the number of memory blocks, K, is selected properly [8, 9] according to the numbers of input and output ports and such that

$$M + N \leqslant NK. \qquad (1)$$

The fabric can also be implemented in S parallel bit-slices to support even higher throughputs and to provide fault tolerance. The more memory bit-slices used, the smaller the word width needed and the higher the aggregate rate that can be achieved assuming a certain access rate for the memory bit-slices [9]. More importantly, recent advances in memory technology [10] have provided impressively high access rates as well as very wide memory words leading to very high aggregate rates and rendering the access rate problem of the shared memory architectures resolved. All of those factors lead to a recent surge of interest in shared memory architectures.

Cells are the fixed size data units used in Asynchronous Transfer Mode, *ATM*, networks and switches while packets are the variable size data units for IP networks. IP/ATM switching has been recently proposed to switch the IP traffic, connection-less variable size packets, over ATM networks, using connection-oriented fixed size cells, to provide faster service with prespecified classes of quality of service, QoS, rather than the best effort service provided by TCP/IP networks. In most implementations of this switching scheme, certain IP packet flows, i.e., persisting connections, are chosen for switching over the ATM network rather than being routed through regular store-and-forward routers of the TCP/IP network. Each packet of those flows is divided into as many fixed size cells as needed and a certain bit in the ATM cell header is used to indicate the end-of-packet cell. Those cells can then be switched through the ATM network as usual. For IP/ATM switching, the design of the switching fabric is exactly the same as that for ATM switches in terms of its logical and physical structure. They only differ in the traffic sources and how their statistics affect the design.

A. The Traffic Source Model

Before we can discuss the power dissipation, a model for the traffic source has to be established in order to be able to derive the transition activity for different nodes throughout the architecture. To that end, the *Discrete-time Batch Markovian Arrival Process* (D-BMAP) [11] is a very flexible model that can be used to model both native ATM traffic classes and IP traffic or an aggregate of them. Formally, a D-BMAP can be defined as a two-dimensional discrete-time Markov process $\{A(k), P(k) : k \geqslant 0\}$ on the state space $\{(i,j) : i \geqslant 0, 1 \leqslant j \leqslant m\}$, where m is the number of states in the irreducible modulating Markov chain, with the transition matrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{D}_2 & \mathbf{D}_3 & \cdots \\ \mathbf{0} & \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{D}_2 & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_0 & \mathbf{D}_1 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \cdots \end{pmatrix}. \qquad (2)$$

The two sets of random variables $\{A(k) : k \geqslant 0\}$ and $\{P(k) : k \geqslant 0\}$ represent the number of cell arrivals on a time slot-per-slot basis, and the state, or the phase, of the underlying Markov chain respectively. Transitions between

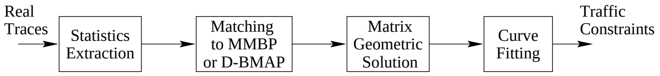


Fig. 2. Traffic Model Development Flow.

subsequent states are governed by the transition matrices $\mathbf{D}_n, n \geqslant 0$, whose elements $(d_n)_{u,v}, 1 \leqslant u, v \leqslant m$, represent the probability that n cells are generated and that the modulating Markov chain is in state v during a slot, given that it was in state u in the previous slot. The matrix $\mathbf{D} = \sum_{n=0}^{\infty} \mathbf{D}_n$ is the transition matrix of the underlying Markov chain, with a stationary probability vector $\boldsymbol{\pi}$ satisfying

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{D}, \quad \boldsymbol{\pi}\mathbf{e} = \mathbf{1}, \tag{3}$$

where $\mathbf{e}$ is a column vector of appropriate size with all elements equal to 1.

The D-BMAP model has the advantage of representing a point process which can be used as input process independently of the queueing system unlike other models such as the fluid approximation method. It also leads to accurate conservative approximations for a large range of system parameters, whereas the fluid approximation is valid only for systems with very large buffer sizes and the M/D/1[2] approximation is valid only for systems with small buffer sizes. Also, the superposition of two D-BMAP processes with m_1 and m_2 states and maximum batch sizes of n_1 and n_2 is also a D-BMAP process with $m_1 \times m_2$ states and $n_1 + n_2$ maximum batch size. This property is very helpful in modeling several different sources separately and aggregating them into one model. Finally, the output process of the system, which is the input process of the next network element, e.g., a multiplexer or a switch, can be modeled again as a Markovian arrival process of the same class as the input process. On the other hand, the number of the states of the Markov chain may grow rapidly, and the use of the special structure of the transition matrix is essential to keep the method computationally tractable for realistic problems. In this paper, the statistics of both ATM traffic and IP flow traces are used to derive the transition matrices of the D-BMAP model using a variation of the matching procedure proposed in [12] and [13]. The statistics of each traffic source are matched to a Markov modulated Bernoulli process, MMBP, which is the special case of a two-state D-BMAP model. This choice mitigates the problem of rapid state growth to a large extent[3]. Finally, the traffic source models are superposed into one D-BMAP model. This flow is shown in figure 2 and elaborated in the next section.

There are so many important traffic models proposed in the literature for different applications and different scenarios [14,15]. For example, self-similar traffic models were originally developed for Ethernet LAN traffic [16]. It would be interesting to investigate the integration of such traffic models here. These types of models can be integrated by representing them as point processes. Moreover, numerical iterative solutions of such models can also be adopted instead of the D-BMAP model although this is not investigated any further in this paper.

B. The Switch Queueing Model

To model the shared memory architecture, a single output multiplexer is considered along with its associated logical queue. The input process is modeled as a D-BMAP process as discussed above. The overall system can then be modeled as a D-BMAP/D/1/B queueing system where the service time is equal to one time slot and the buffering capacity is finite and is equal to B ATM cells. This model can be solved at steady-state to obtain the occupancy probability of the queue. Assuming that the traffic destined to different output ports is independent, the occupancy probability of the shared buffer for a switch with N output ports can be obtained from the N-fold convolution of the occupancy probability for the queue of a single port multiplexer. This probability is the key to our formulations throughout the rest of this paper.

Several methods to derive the steady-state distribution of the buffer occupancy have been proposed in the literature. An efficient numerical method to compute the steady-state probability and the cell loss probability is described in [11] and is used here. This method is based on the Matrix-Geometric solution approach used to solve M/G/1 queueing systems. The Matrix-Geometric algorithm used is described in appendix A. However, the numerical solution is iterative by nature and no closed form solution can be obtained. For example, a specific value for the cell loss probability, CLP, may be used to determine a proper buffer size per port. Formulating the system-level parameter selection as an optimization problem is now complicated by this iterative solution and will require very long solution times. In order to avoid that, a family of curves can be fit to the CLP solution and used in the optimization problem. The CLP is computed using the Matrix-Geometric method iteratively for different switch sizes as a family of curves, such as that shown in figure 3. Real IP flows were used to develop the D-BMAP model used to generate this figure. It is obvious that these curves are almost linear on a logarithmic scale within the range of interest of the buffer size, B, where larger buffer sizes absorb enough burstiness to reduce the cell loss probabilities. Consequently, a linear equation can be fit to these curves as follows

$$\log_{10}(CLP) \geqslant f_1(x, \mathbf{H})B + f_2(x, \mathbf{H}), \tag{4}$$

$$x = \log_2 N, \quad x \in \mathbb{Z}, \quad x \geqslant 1. \tag{5}$$

The choice of system-level parameters does not affect the traffic model and for every design problem the aggregate

[2] A queueing system is concisely described by the *Kendall notation*. This is in the form $A/B/c/K$, where A describes the arrival process, B the service time distribution, c the number of servers, and K the system maximum capacity. If K is omitted, the capacity is infinite. The symbols for A or B are standard where M stands for Markovian and D for deterministic.

[3] For example, modeling 5 multiplexed traffic sources using MMBP results in an overall D-BMAP model of $2^5 = 32$ states. On the other hand, modeling the same 5 sources using 3-state D-BMAP models results in an overall D-BMAP model of $3^5 = 243$ states. This is almost 8 times the number of states and with no significant increase in accuracy. Even worse, 4-state models would require 32 times the numbers of states and so on.

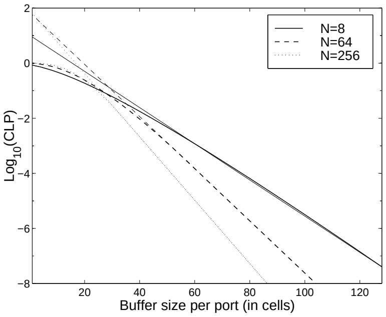


Fig. 3. Sample results from the solution of the D-BMAP modeled traffic and their linear approximations on a logarithmic scale.

traffic pattern expected is assumed to be known. Hence, the transition matrix $\mathbf{H}$ is dependent only on the types of traffic streams and their intensities and is constant for every specific design problem. Consequently, the coefficient functions are reduced to $f_1(x)$ and $f_2(x)$. These two functions can then be approximated by using the least squares regression method to fit them to appropriate equations of the form

$$f_i(x) \approx t_{i,0} + t_{i,1}x + t_{i,2}x^2 + \ldots + t_{i,l}x^l, \quad i \in 1,2 \qquad (6)$$

where $t_{i,j}$ are dependent on the traffic model only and l is an appropriate order to provide enough accuracy and facilitate the optimization process later. All of the traffic models developed for this paper used $l \leqslant 3$. This choice was based on iterative fitting trials where the polynomial order is increased in every iteration and the root mean square (RMS) error is evaluated. It was observed that the RMS error decreases significantly up to an order of 3 after which the decrease is less than 8% for all the data used in our experiments. Moreover, using polynomials of order 3 provides accurate approximations with an RMS error less than 3.6%.

III. Power Analysis and Modeling

The average power, P_{avg}, dissipated in digital systems can be divided into four components; namely, dynamic, short-circuit, leakage and static. However, the short-circuit and static power components are normally negligible in CMOS logic gates. Also, the leakage current, and power, is negligible in advanced fabrication technologies. Hence, the dominant power component is the dynamic one and the overall average power for all nodes in the system can be expressed as follows

$$P_{\text{avg}} \approx P_{\text{dynamic}} = \left(\sum_{\forall i} \alpha_{0\to 1,i} C_i V_i \right) V_{\text{dd}} f_{\text{clk}}, \qquad (7)$$

where, for each node i, $\alpha_{0\to 1,i}$ is its activity factor or the average number of power consuming transitions from 0 to 1 per clock cycle, C_i is its capacitance which is the total contribution of the gate, the diffusion and the interconnect capacitances and V_i is its voltage swing which is usually the same as the supply voltage V_{dd} while f_{clk} is the clock frequency. However, it is not practical to evaluate this power at all nodes in large scale systems and a faster technique is needed.

A. Architectural-level Power Analysis

For such large digital designs, states and dynamic events of the architectural components are too many to use gate- or transistor-level abstractions for the power analysis. Architectural-level abstraction is the level of choice for power analysis during system-level design [17]. This abstraction level is based on a library of architectural building blocks or components implemented as parametric modules. These parametric modules are characterized using a power macro-model based on extracted coefficients and power estimates from accurate lower abstraction levels such as the gate- or transistor-levels.

Evidently, the most appropriate models for high-level power modeling in this case are the cumulative macro-models since we have systems built of several components that differ in their architectures and need to be modeled separately. More importantly, cumulative macro-models capture the system behavior over a long time interval and based on the input statistics which is more appropriate in the case of highly stochastic network traffic that can not be completely described using simple measures such as entropy. These cumulative power macro-models encompass several approaches. One approach is based on *gate equivalent counts* [18, 19] where the complexity of the architectural components is specified in terms of the average number of reference logic gates. The component power is then the number of the reference gates times the power of the reference gate derived from a transistor- or circuit-level analysis. The power due to the interconnects is estimated using variants of Rent's rule for local and intermediate interconnects and H-tree for the clock network [20]. The power for a certain memory architecture is used to model the power of the on-chip memory and improve the estimate of the overall power. The main problem with this approach is that it does not take the transition activity of the real input data into account but rather uses a fixed value for it. Another approach, the *power factor approximation* [21], is based on pre-analyzed models for a library of functional modules. The power for each module is determined using a gate-level abstraction and assuming independent *Uniform White Noise*, UWN, data streams at the inputs. The power of the overall architecture is estimated by identifying the blocks needed to build the system given enough system-level specifications and summing the power of all blocks. Although, this approach accounts for the transition activity, the assumption of independent UWN data is not always valid. In [22], Landman *et al.* proposed the *dual bit type*, DBT, data model. It was shown that for a stream of two's complement samples, typical in DSP applications, bits can be classified into two types. The first type includes the least significant bits whose transition activity can be approximated by a UWN model. Bits of this type will be called the UWN bits. On the other hand, the most significant bits, which represent the sign, may have a relatively low or high transition activity because of a positive or negative temporal correlation between the data samples respectively. These bits represent the second type and will be called the correlated bits. The power estimation in this approach is based on a black-box effective switching capacitance concept. For the bits with UWN activity, the effective capacitance is computed for each and every component using low-level simulation[4] to find the average power, P_o, at reference supply voltage, V_o, and reference frequency, f_o, and using a UWN excitation:

$$C_{uwn} = \frac{P_o}{V_o^2 f_o}. \tag{8}$$

This effective capacitance is then used to estimate the power for a typical bit of the UWN bits at different frequencies and supply voltages. For the correlated bits, the effective capacitance can take one of four values according to the bit transition between successive data samples: $C_t,\ t \in \{00, 01, 11, 10\}$. These values are computed in a similar way as C_{uwn} with the appropriate excitation. If $p_t,\ t \in \{00, 01, 11, 10\}$ were the probabilities of the corresponding bit transitions in the input data stream, then the effective switching capacitance for all bits can be computed using

$$C_T = L_{\mathrm{u}} C_{uwn} + L_{\mathrm{c}} \left(\sum_{t \in \binom{00,01,}{11,10}} p_t \; C_t \right), \tag{9}$$

where L_{u} and L_{c} are the numbers of UWN and correlated bits respectively. Hence, the total power is given by

$$P = C_T V^2 f. \tag{10}$$

Although this technique is limited by the assumption of a specific numeric data representation scheme and by its dependence on the input transitions only, it can be improved to be robust and accurate enough for system-level power analysis. Several successive studies have addressed some of these limitations. They demonstrated the importance of accounting for the transitions at the internal nodes and the use of least mean square regression models to obtain better power estimates [23–26].

Most recently, Bogliolo and Benini [27] proposed another model, the BB model, that is characterization-free for the dynamic power but then uses gate-level simulations to characterize the delay-sensitive second-order power contributions using a regression model. In effect, this is merely modeling the error of the characterization-free part and not necessarily the glitches or short-circuit power. A simpler approach would be to use regression for the whole model. It also assumes that the dynamic power of a certain node is proportional to the transition activity derived from the module Boolean expressions with the node capacitance as the proportionality constant. However, the assumption of the capacitance being independent of the input streams is not always true. For example, the capacitance seen at the inputs of a complex gate differs with the input pattern depending on which transistors are on. The model, however, illustrates other important facts. It shows that not all of the internal node capacitances need to be considered. These node capacitances can be sampled to speed up the characterization with only minor loss of accuracy [23, 27]. It also shows that a uniform random sampling gives the best results. This technique will be used to modify the DBT model as will be described below. Moreover, all of these schemes still need to be generalized to accommodate different types of data representation such as that for network traffic streams commonly found at the input ports of the switching fabrics.

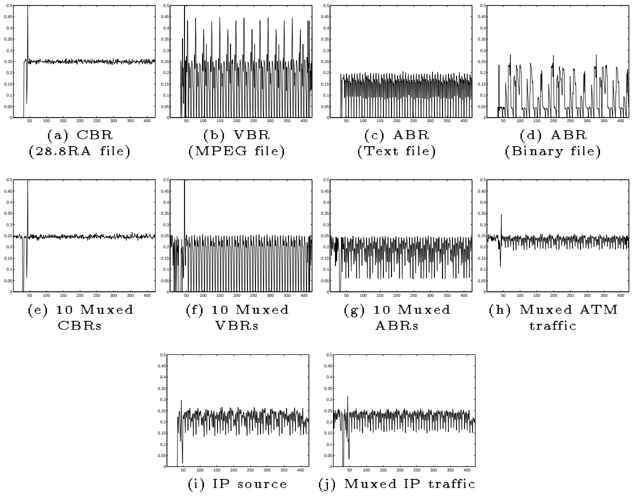

(a) CBR (28.8RA file)
(b) VBR (MPEG file)
(c) ABR (Text file)
(d) ABR (Binary file)
(e) 10 Muxed CBRs
(f) 10 Muxed VBRs
(g) 10 Muxed ABRs
(h) Muxed ATM traffic
(i) IP source
(j) Muxed IP traffic

Fig. 4. Transition activity factor vs. bit number in an ATM cell for various traffic streams.

B. Bit-level Analysis

Before studying the transition activity for each bit in an ATM cell, we need to point out the fact that all of the components used in the switching fabric discussed have no activity coupling between adjacent bits and it is safe to assume independent distributions for the transition activity of the cell bits. Thus, it is necessary to study the activity factor at the bit-level for all of those components when they are excited with real network traffic streams. This was done for several types of ATM traffic streams for all 424 bits of the ATM cell. Single streams at the UNI as well as multiplexed streams at the NNI were used. These included CBR, VBR and ABR traffic processed using AAL1, AAL2 and AAL5 layer respectively[5]. Moreover, multiplexed ATM traffic from all three classes, single and

[4] Power annotated gate-level simulation, IRSIM or POWERMILL switch-level simulation or SPICE circuit-level simulation whenever possible.

[5] In ATM terminology, UNI stands for User-Network Interface or the edge of the network where the users access it while NNI stands for the Network-Network Interface between the different switching nodes in the core of the network. Also, ATM traffic has several classes depending on the application and the quality of service it requires. These classes include the constant bit rate (CBR), the variable bit rate (VBR) and the available bit rate (ABR) among others. Moreover, the standards define several ATM adaptation layers, such as AAL1, AAL2 and AAL5, that should be used with certain classes of traffic to prepare or adapt the application traffic to the actual ATM cells that will carry it.

multiplexed IP packet streams over ATM cells were also analyzed. Typical results are shown in figure 4.

From these results, it is obvious that a generalized form of the DBT model [22] is needed. Depending on the traffic mix, the bits of the ATM cell are decomposed into activity regions. Bit regions with activity factor approaching 0.25 are assumed to have UWN activity. Each of the remaining correlated regions is similar to the sign bits in DSP applications except that each region has different transition probabilities. The probabilities of bit transitions t estimated for each region r are $p_{t,r}$, $t \in \{00, 01, 11, 10\}$. L_u and $L_{\text{c},r}$ are the total number of bits in the UWN regions and the number of bits in each correlated region r. Consequently, the effective switching capacitance is given by

$$C_T = \frac{L_\text{u}}{L/S}[\mathbf{C}_\text{eff}.\mathbf{L}_\text{f}] + \sum_{\forall r} \frac{L_{\text{c},r}}{L/S}\left[\sum_{t \in \binom{00,01,}{11,10}} p_{t,r}[\mathbf{C}_t.\mathbf{L}_\text{f}]\right], \tag{11}$$

where $\mathbf{C}_\text{eff}$ and $\mathbf{C}_t$ represent the effective capacitance vector of a bit-slice of the design for UWN bits and for the correlated bits with transition t respectively and $\mathbf{L}_\text{f}$ represents the bit-slice complexity vector. To account for all bits, the weights $L_\text{u}/(L/S)$ and $L_{\text{c},r}/(L/S)$ are included, where L is the cell width in all bit-slices and is normally equal to 424 and S is the number of bit-slices.

C. Cell-level Analysis

The capacitance switched is influenced by the arrival rate of cells at each module. This can be derived from the D-BMAP model in section II-A and is given by

$$\rho = \boldsymbol{\pi}\left(\sum_{n=0}^{\infty} n\mathbf{D}_n\right)\mathbf{e}. \tag{12}$$

It is also influenced by the module type and dimensions. The switched capacitance for the static memory blocks has the form

$$C_\text{mem} = (c_3\frac{MB}{K}\frac{L}{S} + c_2\frac{MB}{K} + c_1\frac{L}{S} + c_0)K\rho, \tag{13}$$

where those capacitive terms are computed for the memory bit array, the decoding circuitry, the sense amplifiers and other periphery circuits, and the control circuitry of the memory respectively. The coefficients include the effect of the layout and the fabrication technology parameters. Any power management technique used to deactivate unused blocks will have to be considered in the equation through a probabilistic weight. Similarly, the switched capacitance for the crossbars is given by

$$C_\text{xbar} = c_5(p_xM + p_bN)K\frac{L}{S}\rho + c_4\rho, \tag{14}$$

where p_x and p_b are the probabilities that a crossbar element is in a cross or a bar state respectively and are constants specific to the distribution of source-destination address pairs in the traffic at the ingress crossbar and to the scheduling algorithm at the egress crossbar. The second term in equation (14) is due to the control circuitry of the crossbars. The capacitive coefficients, $\mathbf{C}_\text{eff}$ or $\mathbf{C}_t$, and the complexity parameters, $\mathbf{L}_\text{f}$, can then be written in vector form as follows

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix}, \tag{15}$$

$$\mathbf{L}_\text{f} = \rho\begin{bmatrix} K & K\frac{L}{S} & MB & MB\frac{L}{S} & 1 & (p_xM + p_bN)K\frac{L}{S} \end{bmatrix}^T. \tag{16}$$

C.1 Extracting the Capacitive Coefficients

To extract the capacitive coefficients several steps have to be followed. An input stream or pattern has to be generated using a certain value of input temporal correlation for every parametric module we have. This pattern is then used to simulate the modules using a switch-level or transistor-level simulator with capacitance measurement capabilities to estimate the average capacitance. The use of switch-level or transistor-level simulation takes into consideration the glitching power because of the inherent inclusion of a delay model in these simulations. In [23, 27], it was shown that, for a given module, the energy for a particular transition from one input to the next is proportional to the energy estimated using only a uniform random sample of internal nodes. During this simulation step, the internal nodes are also sampled to extract their corresponding capacitances which depend on the input transition propagated through the circuit. The data generated in this step represents equation (15). However, this data must be fit to the regression model that represents the complexity of the module. To do this, the simulation is repeated for each parametric module over a wide range of parameters and the resulting capacitances are fit using the least square fitting technique. The whole process is then repeated using different correlations for the input pattern to extract separate capacitance coefficients for the different correlations, i.e., C_t, $t \in \{uwn, 00, 01, 11, 10\}$. In short, this model uses general bit types for the different transition activity regions and models them using sampled capacitances in a regression power macro-model. Hence, we will refer to it as the *General Sampled-Bit Regression* model (GSBR).

D. Numerical Evaluation of the Model

To check the robustness and accuracy of this model, the different components along with 2 × 2, 4 × 4 and 8 × 8 test fabrics were built in a 0.8μm BiCMOS, 0.5μm CMOS and 0.35μm CMOS technologies. These fabrics were merely for test purposes and each had a buffer space that guarantees a cell loss probability less than or equal to 10^{-5}, which is a bit relaxed, for all the D-BMAP models used. To develop those D-BMAP models, multiplexed IP streams captured from the core of the network, or the NNI, were divided into two sets and used to derive the traffic models $\text{IP}_{c,1}$ and $\text{IP}_{c,2}$. Similarly, multiplexed ATM streams captured from the core were used to develop models $\text{ATM}_{c,1}$ and $\text{ATM}_{c,2}$ while those captured at the edge, or the UNI, were used to develop $\text{ATM}_{e,1}$.

Using the $\text{IP}_{c,1}$ model and the three different fabrication technologies, the power dissipation estimated using the

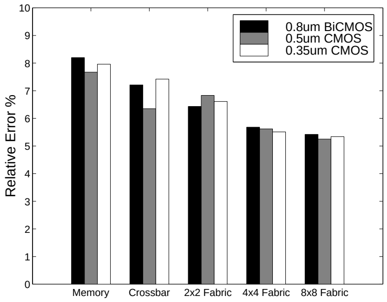


Fig. 5. Relative errors for the power dissipation estimates using different fabrication technologies compared to the evaluated values (fixing the $\mathrm{IP}_{c,1}$ traffic model).

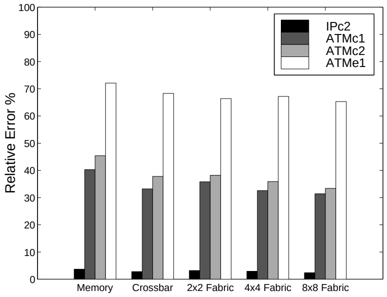


Fig. 6. Relative errors for the power dissipation estimates using the $\mathrm{IP}_{c,1}$ traffic model compared to the estimates using other traffic models (fixing 0.35μm CMOS technology).

proposed power model was compared to the power evaluated using low-level simulations. Figure 5 shows that the relative errors did not exceed 8.4% for any of the tested components. These results show that the power model is robust and accurate enough to be used in system-level design. Moreover, the errors seem to be decreasing with the size of the test fabrics which makes the model more suitable as larger parameter values are explored.

Also, those same test components implemented using the 0.35μm CMOS technology were used to compare the power dissipation estimated for the $\mathrm{IP}_{c,1}$ model to that estimated for the other traffic models. The relative errors shown in figure 6 indicate that the power model of the fabric is indeed dependent on its input traffic. This is evident from the large errors associated with traffic types other than the IP traffic used to estimate the power dissipation. Moreover, the figure shows that the only acceptable relative error is the one for the $\mathrm{IP}_{c,2}$ model. These results assert the accuracy of the D-BMAP model in modeling specific aggregates of traffic and its suitability for use with the power model.

In order to compare the performance of the proposed model relative to the other macro-models, the power estimates are compared for the same $\mathrm{IP}_{c,1}$ traffic model and the 0.5μm CMOS technology using the proposed model (GSBR), the dual bit type model (DBT) and the Bogliolo and Benini model (BB). Uniform sampling was used to select the nodes in both the GSBR and the BB models. A 50% node sampling ratio was used for both models. This ratio was used to reasonably speedup the GSBR characterization at the expense of small RMS relative errors of less than 3%. The sampling was also constrained to guarantee coverage of all available transition activity regions. The errors compared in figure 7 are again the relative errors of the estimated power using these models to the power evaluated using switch-level simulations [28]. The results show that the proposed model improves over the DBT model considerably. An interesting point is that the DBT model fares better for memory than the BB model because the DBT model takes into consideration the detailed blocks of the memory while the BB model assumes the dependency on the input transitions is only manifested in the logic expressions which is not completely true for memories and it also ignores the dependency of the capacitance on the input transitions. On the other hand, the proposed model acknowledges the dependency of the capacitance on the input transitions, by depending on transistor-level baseline simulations, and improves its modeling using the least mean squares regression similar to the DBT model. Moreover, it recognizes the different types of bit transition models rather than only two types. It also uses the node sampling concept to speed up the characterization without any significant loss of accuracy. It does not, however, employ the other features of the BB model such as being based on logic equations or providing instantaneous power estimates independently from the applied input pattern since these features defeat the purpose of the traffic-driven optimization framework.

IV. Other System-Level Requirements

A. Area Requirements

Heuristic formulas similar to those used in [7] were derived using parameterized deep submicron layouts for the area of the crossbar elements, A_{xbar}, and the static memory cells, A_{cells}. On the other hand, parametric VHDL models and layouts for primitive logic gates were used to model other decoding and periphery circuits, A_{prphry}. These formulas are:

$$A_{\mathrm{xbar}} = a_0\lambda^2 + a_1\lambda^2(M+N)K\frac{L}{S}, \tag{17}$$

$$A_{\mathrm{cells}} = a_2\lambda^2 MB\frac{L}{S}, \tag{18}$$

$$A_{\mathrm{prphry}} = \lambda^2 K\left(a_3 + a_4\frac{MB}{K} + a_5\frac{L}{S}\right), \tag{19}$$

where λ is the characteristic length for the technology to be used. Interconnect area was accounted for in the above equations by using layout measurements of interconnect length or estimates from Rent's rule. The above procedure was repeated for the three technologies available and the scaling factors, $a_i, i = 0, 1, \ldots, 5$, were found using the least squares method. Consequently, the total area of a single bit-slice is given by

$$\begin{aligned} A = \lambda^2\Big((a_0 + a_3K) + (a_4 + a_2\frac{L}{S})MB \\ + (a_5K + a_1(M+N)K)\frac{L}{S}\Big). \end{aligned} \tag{20}$$

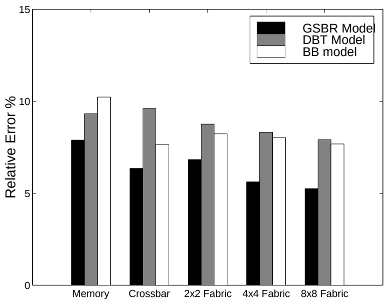


Fig. 7. Relative errors for power dissipation estimates for different power macro-models (using the $\mathrm{IP}_{c,1}$ traffic model and the 0.5μm CMOS technology).

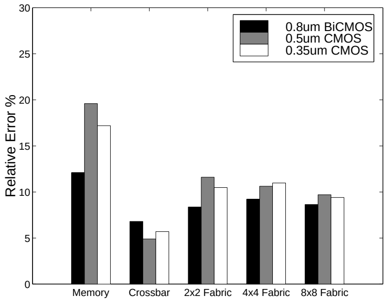


Fig. 8. Relative errors for the area estimates compared to the actual layout area for different fabrication technologies.

Again, this area should be constrained for every chip depending on the yield, packaging type and the active area of the die. An empirical relation similar to that in [7] will be used here

$$A < 1.35 \times 10^{-4}\ \lambda^{-0.85}. \tag{21}$$

Figure 8 shows the relative errors for the area estimated using these heuristics. The errors are within 19.6% which is acceptable considering the fact that the layout and the physical design rules may differ widely between different technologies. If more design rules and technology parameters were included, the formulation would be too complicated and the accuracy gained would not be significant for a system-level constraint.

B. I/O Pin Constraints

In a switching fabric design, many constraints can be imposed due to the I/O pins used [7]. The pin count can limit the number of chips used and can even impose a certain type of packaging. On the other hand, the pin count itself can be limited by the chip area and the I/O pad size and pitch. Pin count can also limit the aggregate throughput due to the pin speed limits. A constraint that is more related to this work is the I/O power due to the pads associated with those pins. At very high link rates, complex analog circuitry might be needed and should be included in the analysis. However, the architecture proposed here uses parallel buses over several bit-slices to reduce the rate required per I/O pin. This is practical and appropriate for applications such as satellite and access switches. Consequently, traditional low-voltage CMOS I/O implementations can be used and a similar analysis can be applied. The switched capacitance for the pads would then be given by

$$C_{\mathrm{PADS}} = (c_6 M \frac{L}{S}\rho + c_7 N \frac{L}{S}\rho_o + c_8), \tag{22}$$

where ρ_o is the cells departure rate from the fabric. It was shown in [11] that the departure process of a D-BMAP/D/1/B queue is a D-MAP process which is a special case of the D-BMAP process. The departure process can be derived from the queueing model and the departure rate ρ_o can then be calculated. The first two terms in equation (22) represent the switching capacitance for the input and output pads respectively. The last term represents the remaining control pads. These capacitive coefficients and complexity parameters can be appended to the vectors in equations (15) and (16) to add the power dissipation of the pins to the fabric power expression.

C. Other Performance Constraints

The aggregate traffic rate, or throughput, expected from an $M \times N$ fabric with a link rate of V cells/sec is limited by the number of bit-slices used and by the access rate, or bandwidth, of the memory used, $\mathrm{BW}_{\mathrm{mem}}$, according to the following relation

$$S \geqslant \frac{NVL}{\mathrm{BW}_{\mathrm{mem}}}. \tag{23}$$

Depending on the application that the switching fabric is being designed for, equations governing the cell transfer delay, CTD, or the cell delay variation, CDV, can be formulated similar to those for CLP and included as constraints for the optimization problem if necessary.

V. Optimization for Low-Power

The proposed framework formulates and solves the design problem as an optimization problem. The main objective of this framework would be to minimize the power dissipation within the fabric. Other applications may regard the chip area or some other requirement as the objective function. Multi-objective optimization can also be used to minimize the power dissipation along with other requirements. In global satellite networks, a limited number of ports is needed and the shared memory switching fabric can be used as a stand-alone fabric. M is commonly taken to be equal to N to simplify the design problem. Consequently, several equations are simplified. The constraint on K in equation (1) will be

$$K \geqslant 2. \tag{24}$$

Similarly, the complexity vector, the total area and the switched capacitance for the pads in equations (16), (20) and (22), respectively, can be rewritten as

$$\mathbf{L_f} = \rho \left[K \quad K\tfrac{L}{S} \quad NB \quad NB\tfrac{L}{S} \quad 1 \quad 2(p_x + p_b)NK\tfrac{L}{S}\right]^T, \tag{25}$$

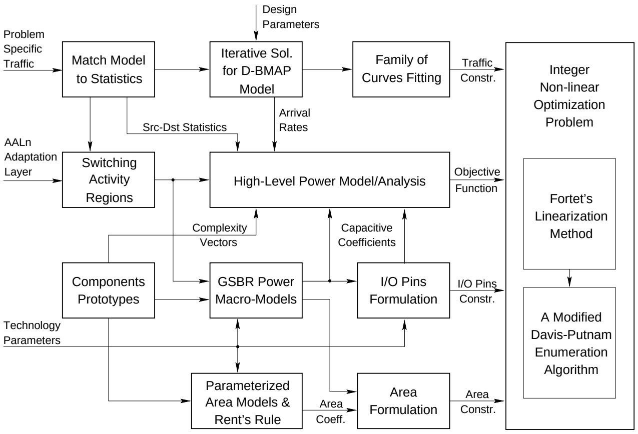


Fig. 9. Switching fabric design in satellite applications at the system-level formulated as an integer non-linear optimization problem.

$$A = \lambda^2\Bigg((a_0 + a_3K) + (a_4 + a_2\frac{L}{S})NB + (a_5K + 2a_1NK)\frac{L}{S}\Bigg) \tag{26}$$

$$\text{and} \quad C_{\text{PADS}} = (c_6N\frac{L}{S}\rho + c_7N\frac{L}{S}\rho_o + c_8). \tag{27}$$

The design problem is formulated as an integer non-linear optimization problem since the design parameters, i.e., N, B, K and S, are all integers. Figure 9 summarizes how the models are derived and how they interact within the proposed framework. The objective is to minimize equation (10), with C_T defined by equations (11), (15), (25) and (27). The different constraints define the solution domain for the optimization problem. Equations (4) and (5) define the performance constraints while equations (23) and (24) set lower bounds for the numbers of slices and shared memory blocks respectively. The area constraints are defined by equations (21) and (26).

Integer non-linear optimization problems are difficult to solve. However, this problem can be solved in two steps. The first step uses Fortet's linearization method to linearize the problem which introduces new variables but maintains a polynomial size for the problem. Then, an implicit enumeration algorithm based on Davis-Putnam enumeration algorithm, which is a logic-based branch-and-cut optimization algorithm, is used to find the optimal solution [29,30]. This method requires the enumeration of all allowed values of the variables as separate Boolean variables. Although this requires an additional step in the problem formulation, it allows for speeding up the optimization algorithm by enumerating only those variable values that are valid feasible solutions and ignoring all other invalid values, i.e., the algorithm will not search through invalid design space regions, if there are any, to save time. Moreover, the accuracy of the solution is affected by the linearization process, however, it is still accurate enough for system-level design. This algorithm is also fast enough to explore many regions of the design space and to make informed system-level decisions. Table I lists the results for several fabric optimization experiments with different traffic models and fabrication technologies. The power dissipations obtained in these experiments are considerably below the values reported in [7] for comparable fabrics. Also, variations in the problem constraints were used, such as adding a lower bound of 32 for the number of ports in experiment number 6 and limiting the pin count per chip to 300 in experiment number 7. It should also be noted that optimizing the components of the fabric at lower levels of abstraction may further reduce the power dissipation.

TABLE I

Experimental results for low-power fabric design.

Exp. #	Traffic Model	CLP	Tech. λ μm	Aggr. Mem. Rate Gb/s	Link Rate Gb/s	N	B	K	S	Pwr. Est. (W)
1	$\text{IP}_{c,1}$	10^{-10}	0.8	3.2	2.4	8	1024	4	10	11.67
2	$\text{IP}_{c,2}$	10^{-10}	0.5	3.2	2.4	16	1024	4	14	9.98
3	$\text{ATM}_{c,1}$	10^{-10}	0.5	3.2	9.6	4	2048	6	12	21.82
4	$\text{ATM}_{c,2}$	10^{-12}	0.35	3.2	2.4	16	2048	6	12	11.47
5	$\text{ATM}_{c,2}$	10^{-12}	0.35	4.8	2.4	32	2048	8	18	24.66
6	$\text{ATM}_{c,2}$	10^{-10}	0.35	3.2	0.622	128	2048	12	24	22.84
7	$\text{ATM}_{c,2}$	10^{-10}	0.35	3.2	0.622	32	1024	2	8	2.71

As mentioned before, typical cell loss probabilities for satellite applications are 10^{-10} or less because retransmissions due to errors or cell loss can not be tolerated due to the large propagation delay. Consequently, validating the D-BMAP traffic model using simulations is almost impossible for low cell loss probabilities because the simulation time grows exponentially as the absolute value of the exponent of the cell loss probability grows linearly. The best we can do is to simulate the system under that traffic model up to a certain feasible value of the cell loss probability and then extrapolate the results for lower cell loss probabilities. Another useful approach that can help to validate the overall effect of the traffic model on the design framework is to test for the sensitivity of the system-level framework to variations in the D-BMAP model parameters used to develop the performance constraints. To this end, we have conducted experiments number 8 through 11 in table II which were basically repetitions of experiments 2 and 4 except that the parameters of the traffic model were perturbed by a factor of +5% in experiments 8 and 10 and by a factor of -5% in experiments 9 and 11 as compared to the actual values in experiments 2 and 4 respectively. The results vary somewhat in terms of the estimated power dissipation values. However, they show a high degree of robustness in terms of the fabric complexity parameters which indicates that the overall framework is not largely susceptible to variations in the traffic model. Experiments number 12 through 15 are similar except that now the traffic models are perturbed by +10% and -10%. At such high perturbations, the power dissipation variations are still within 21% depending on the traffic model used. However, the fabric complexity parameters are now more susceptible to these perturbations. In general, this validation shows that the traffic model is not that critical in terms of the accuracy and robustness of the overall framework particularly when used as a system-level tool. These results also show that

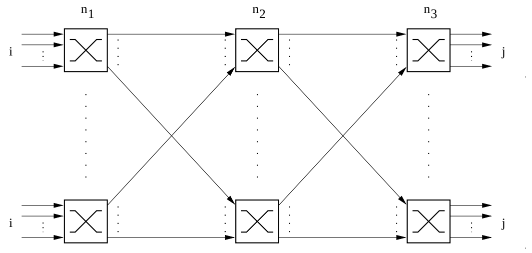


Fig. 10. General configuration of $\mathcal{C}(i, j, n_1, n_2, n_3)$ Clos network.

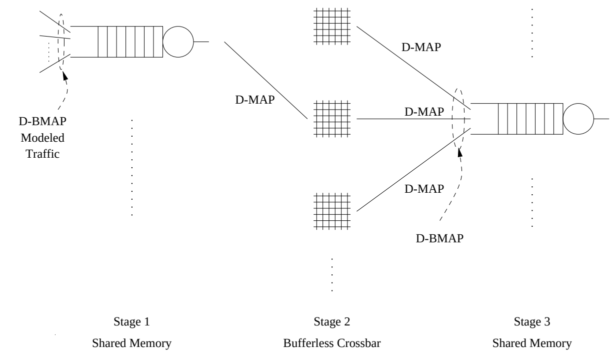


Fig. 11. The traffic models used to define the performance constraints through the stages of a Clos network.

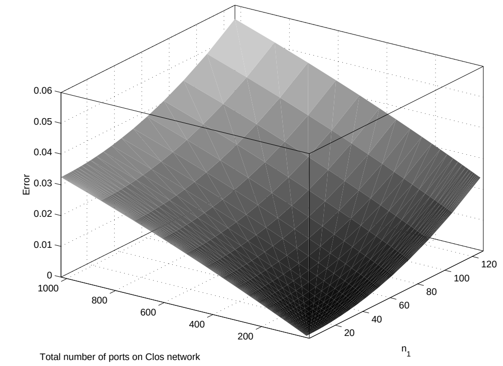


Fig. 12. Relative error for the power estimates using the shared memory modules framework as compared to a detailed formulation framework (The error surface is fit to the sampled points).

the traffic model chosen represents the different types of traffic accurately enough for the overall framework to be used properly to study the system-level alternatives.

To evaluate the accuracy of the power estimates provided by the framework, actual low-level simulations have to be used to provide the reference power dissipation estimates. Unfortunately, this is computationally expensive and can not be used for all of the large fabrics used in our experiments. To get a feeling of this accuracy, however, experiment number 1 was simulated and the power estimate from the framework had an error of 14.36%. On the other hand, the simulation run-time was more than 60 times that of the optimization run-time and we expect it to be even much higher for the larger fabrics. Although, one value can not be used to draw credible conclusions with regards to the framework accuracy, it still shows that it can be relied on for system-level and design space explorations.

A. Framework Extension for Scalability

Different methods can be considered to extend this framework to accommodate larger scalable switching fabrics. In this section, we will use a strictly non-blocking Clos network, SNB $\mathcal{C}(i, j, n_1, n_2, n_3)$, as shown in figure 10. In this case, shared buffer fabrics are used as modules for the first and third stage while bufferless crossbars are used as the modules in the second stage to maintain the traffic cell order. Other networks could have been used depending on the system requirements and architecture. We further assume equal number of modules in both the first and third stages, $n_1 = n_3$, and equal number of inputs to the first stage and outputs of the third stage, $i = j$. One method to extend the optimization framework would be to reformulate all of the detailed equations of the power macro-model and the heuristic equations for the silicon area to take into account all the modules in the Clos network. However, this formulation will add more variables, equations and iterations to many parts of the framework requiring excessively longer times to find different solutions and to explore the design space which defeats the whole purpose of this work. To avoid that, the same framework for single shared memory $M \times N$ fabric modules is used with minor modifications. Note that both M and N are still treated as optimization parameters. First, we assume that the required number of input or output ports, $\mathcal{N}$, for the scalable fabric is known and constant. With the assumption that $i = j = M$, the number of $M \times N$ modules in the first stage and $N \times M$ modules in the third stage can be found as

$$n_1 = n_3 = \frac{\mathcal{N}}{M} \tag{28}$$

The number of the crossbar modules in the second stage is simply equal to the number of the output ports of one shared memory module, N. Next, we need to add another constraint to account for the non-blocking condition as given for SNB Clos networks by the following equation

$$n_2 \geqslant i + j - 1. \tag{29}$$

With the above assumptions, this condition reduces to the following

$$N \geqslant 2M - 1. \tag{30}$$

Minor modifications can also be done at this stage to accommodate the requirements of fault tolerance by changing the complexity variables to reflect the redundancies needed.

The formulation of the performance constraints in this case is also a bit different. Figure 11 shows the different traffic models through the stages of a Clos network. A D-BMAP model is used to model the aggregate sources for a

TABLE II

Experimental results for low-power fabric design using perturbed traffic models.

					Tech.	Aggregate	Link					Pwr.
Exp.	Original	Traffic	Perturb.		λ	Mem. Rate	Rate					Est.
#	Exp. #	Model	%	CLP	μm	Gb/s	Gb/s	N	B	K	S	(W)
8	2	$\mathrm{IP}_{c,2}$	+5%	10^{-10}	0.5	3.2	2.4	16	1024	5	14	10.76
9	4	$\mathrm{ATM}_{c,2}$	+5%	10^{-12}	0.35	3.2	2.4	16	2048	7	12	13.16
10	2	$\mathrm{IP}_{c,2}$	-5%	10^{-10}	0.5	3.2	2.4	16	1024	4	14	9.21
11	4	$\mathrm{ATM}_{c,2}$	-5%	10^{-12}	0.35	3.2	2.4	16	2048	6	12	9.89
12	2	$\mathrm{IP}_{c,2}$	+10%	10^{-10}	0.5	3.2	2.4	16	1024	7	14	11.28
13	4	$\mathrm{ATM}_{c,2}$	+10%	10^{-12}	0.35	3.2	2.4	16	2048	7	14	13.83
14	2	$\mathrm{IP}_{c,2}$	-10%	10^{-10}	0.5	3.2	2.4	16	1024	5	12	8.92
15	4	$\mathrm{ATM}_{c,2}$	-10%	10^{-12}	0.35	3.2	2.4	16	2048	5	12	9.36

TABLE III

Experimental results for scalable low-power fabric design using $\lambda = 0.35\mu$M, CLP $= 10^{-10}$ and $\mathcal{N} = 512$.

		Aggr.							
		Mem.	Link						
Exp.	Traffic	Rate	Rate		$i = j$				
#	Model	Gb/s	Gb/s	$n_1 = n_3$	$= M$	$n_2 = N$	B	K	S
1	$\mathrm{IP}_{c,2}$	3.2	2.4	32	16	32	2048	6	28
2	$\mathrm{ATM}_{c,1}$	3.2	2.4	32	16	32	4096	10	26
3	$\mathrm{ATM}_{c,1}$	4.8	2.4	64	8	16	4096	16	12
4	$\mathrm{ATM}_{c,2}$	3.2	0.622	8	64	128	2048	14	26
5	$\mathrm{ATM}_{c,2}$	3.2	0.622	16	32	64	1024	6	14

certain logical queue at the first stage. CLP and other performance measures can be extracted from the distribution of the buffer occupancy in the shared memory pool which is function of N but not M for this stage. Similarly, the traffic coming out of a first stage output can be modeled using a D-MAP model. This traffic goes through the crossbars and is aggregated into a D-BMAP model as it feeds into the logical queues of the third stage. This time, the performance measures are function of M rather than N. Curves similar to those obtained in figure 3 are used here to derive the performance constraints. For example, a specification given for the overall CLP of the network would have to be larger than the values derived for shared memory modules both with N and M output ports and weighted by the numbers of modules in the first and third stages, n_1 and n_3, respectively. A test case was used to evaluate the error of the estimated power using that method as compared to that using a detailed formulation model. The parameters used were $M = 8, B = 1024, K = 4$ and $S = 10$ and using a link rate of 2.4Gb/s and a 0.5μm CMOS technology. Figure 12 shows a surface fit to the relative error at sampled points and that surface is within approximately 6% for fabrics with $\mathcal{N}$ up to 1024. This error is small enough to justify the use of this method over the computationally expensive detailed formulation. Moreover, this error can be reduced considerably by taking into account the $N \times n_1^2$ crosspoints of the second stage. Table III lists the results for several scalable fabric optimization experiments following the method described above. The time consumed in arriving at any of these results is comparable to that in shared memory module experiments and the framework can still be used to explore the design space for scalable networks.

VI. Conclusions

Low-power system-level design of switching fabrics is becoming a necessity with the emergence of on-board satellite switching and advanced VLSI technologies. This work has proposed a framework for system-level design optimization that provides for quick and efficient exploration of the design space without compromising the design quality. Its power lies in the incorporation of the D-BMAP traffic model with a high-level power macro-model and in matching it to real traffic statistics. Advances in the area of integer non-linear optimization may also increase the efficiency of this framework by providing more accurate or faster solutions for the optimization problem. The results obtained had remarkable power reductions compared to other methods that do not include traffic models and statistics. This also emphasizes the importance of moving the power optimization process from the circuit-level to the system-level. Another significant advantage of this framework is its expandability and accommodation of scalable networks.

Appendix

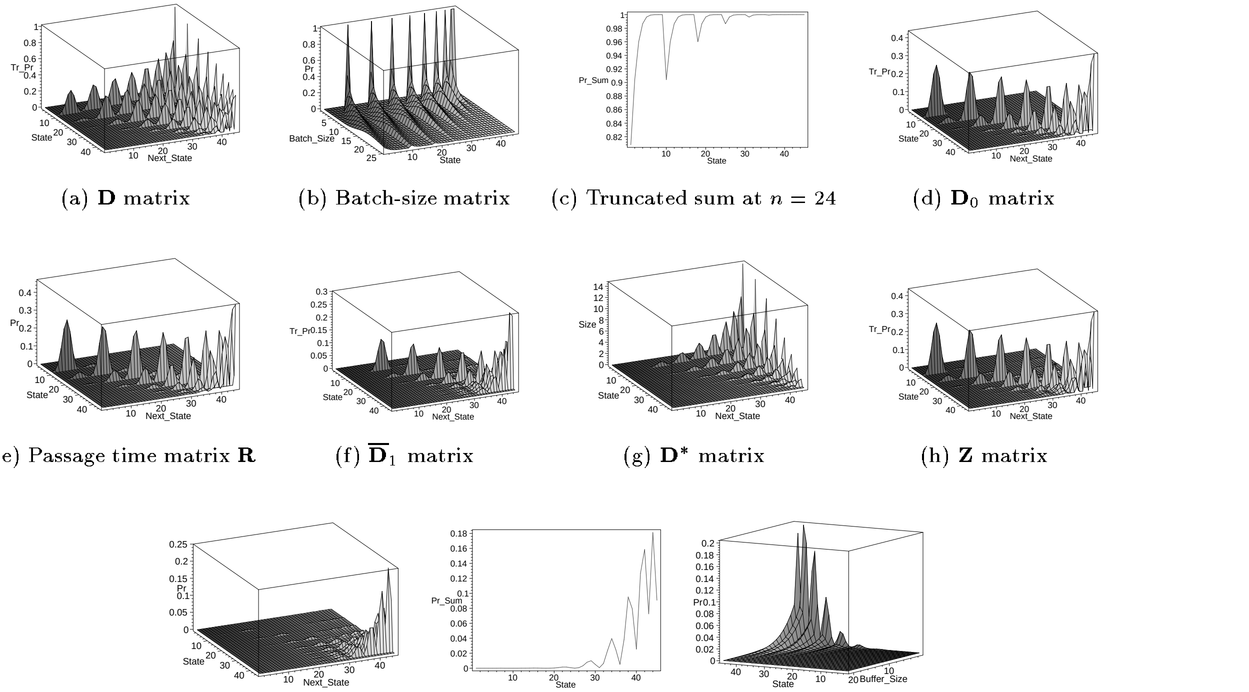
I. Matrix-Geometric Solution Algorithm

In recent years, several methods have been proposed to solve models of M/G/1 type and quasi-birth-death (QBD) models. Li *et al.* [31, 32] proposed a generalized folding algorithm [33]. However, the D-BMAP/D/1 is simple enough and can be solved off-line which makes the matrix-geometric solution algorithm a better candidate for our application. The matrix-geometric solution was first proposed [34, 35] as a generalized analytic technique for the embedded Markov chains of the M/G/1 and GI/M/1 queueing models. An excellent introduction to its application to the batch Markovian arrival process can be found in [36]. The algorithm used here follows the discrete case of [37]. It uses the following steps.

1. Start with a stochastic or zero matrix $\mathbf{R}(0)$ and iterate to get $\mathbf{R}$, also known as the *passage time matrix*:

$$\mathbf{R}(k) = \sum_{n=0}^{N_{trnc}} \mathbf{D}_n[\mathbf{R}(k-1)]^n, \qquad (31)$$

where N_{trnc} is the batch size at which we can truncate computations without loss of accuracy and k is the iteration



(a) $\mathbf{D}$ matrix (b) Batch-size matrix (c) Truncated sum at $n=24$ (d) $\mathbf{D}_0$ matrix

(e) Passage time matrix $\mathbf{R}$ (f) $\overline{\mathbf{D}}_1$ matrix (g) $\mathbf{D}^*$ matrix (h) $\mathbf{Z}$ matrix

(i) $\mathbf{T}$ matrix (j) π vector (k) Buffer occupancy $\mathbf{x}$

Fig. 13. Sample results of the matrix-geometric algorithm solving a D-BMAP model of 8 identical bursty sources with $\rho \approx 1.48$.

number. The iteration process ends when

$$\max_{i,j}\left| R_{i,j}(k) - \{\sum_{n=0}^{N_{trnc}} \mathbf{D}_n[\mathbf{R}(k)]^n\}_{i,j}\right| \leqslant \epsilon \quad \forall i,j, \tag{32}$$

where the subscripts i,j denote the element in row i and column j of the corresponding matrix and ϵ is a very small number around 10^{-10}. Since $\mathbf{D}_n$ is assumed to be 0 for $n > N_{trnc}$, the matrix polynomial $\sum_{n=0}^{N_{trnc}} \mathbf{D}_n\mathbf{R}^n$ can be efficiently evaluated using Horner's rule [34] and at each iteration $\mathbf{R}(k)$ needs to be normalized to be a stochastic matrix.

2. Let $\mathbf{x}(i)$ be a vector of m elements, one for each state of the enumerated states of the D-BMAP model. If this vector represents the probability of having i cells in the buffer or the occupancy probability of the buffer, compute the vector $\mathbf{x}(0)$ as follows:

$$\overline{\mathbf{D}}_i = \sum_{n=0}^{N_{trnc}-i} \mathbf{D}_{i+n}\mathbf{R}^n, \tag{33}$$

$$\mathbf{D}^* = \sum_{n=1}^{N_{trnc}} n\mathbf{D}_n, \tag{34}$$

$$\mathbf{Z} = \mathbf{D}_0 + \overline{\mathbf{D}}_1(\mathbf{I}-\overline{\mathbf{D}}_1)^{-1}\mathbf{D}_0, \tag{35}$$

$$\mathbf{T} = \overline{\mathbf{D}}_1\mathbf{R}, \tag{36}$$

$$\rho = \pi\mathbf{D}^*\mathbf{e} \quad \text{and} \tag{37}$$

$$\mathbf{x}(0) = \frac{1}{d}\mathbf{z}, \tag{38}$$

where $\mathbf{z}$ is the invariant probability vector of $\mathbf{Z}$ satisfying $\mathbf{zZ} = \mathbf{z}, \mathbf{ze} = 1$, and

$$d = 1 + \frac{1}{1-\rho}\mathbf{z}[\mathbf{I} + (\mathbf{D}-\mathbf{D}_0-\mathbf{T})(\mathbf{I}-\mathbf{D}+\mathbf{e}\pi)^{-1}]\mathbf{D}^*\mathbf{e}. \tag{39}$$

3. Compute $\mathbf{x}(i)$ for $i \geqslant 1$ by Ramaswami's recurrence [36]:

$$\mathbf{x}(i) = \Bigg[\mathbf{x}(0)\overline{\mathbf{D}}_i + \sum_{n=1}^{\min(i-1,i-N_{trnc}+1)} \mathbf{x}(n)\overline{\mathbf{D}}_{i-n+1}\Bigg](\mathbf{I}-\overline{\mathbf{D}}_1)^{-1}. \tag{40}$$

The traffic intensity or cell arrival rate given in equation (37) is again equivalent to that given by equation (12) under the assumption that $\mathbf{D}_n = \mathbf{0}$ for $n > N_{trnc}$. Sample experimental results for this algorithm are shown in figure 13. This analysis was done for a D-BMAP model of 8 identical bursty sources with $\rho \approx 1.48$. It had $\frac{(8+1)(8+2)}{2} = 45$ enumerated states representing which and how many sources are on. The buffer occupancy derived can then be used to derive different performance constraints for different designs.

References

[1] F.A. Tobagi, "Fast packet switch architectures for broadband integrated services digital networks," *Proceedings of the IEEE*, vol. 78, no. 1, pp. 133–166, Jan. 1990.

[2] R.Y. Awdeh and H.T. Mouftah, "Survey of ATM switch architectures," *Computer Networks and ISDN Systems*, vol. 27, no. 12, pp. 1567–1613, Nov. 1995.

[3] S. Shaikh, M. Schwartz, and T. Szymanski, "A comparison of the Shuffelnet and the Banyan topologies for broadband packet switches," in *Proc. IEEE INFOCOM*, 1990, pp. 1260–1267.

[4] P. Coppo, M. D'Ambrosio, and R. Melen, "Optimal cost/performance design of ATM switches," in *Proc. IEEE INFOCOM*, May 1992, pp. 446–458.

[5] E.W. Zegura, "Architectures for ATM switching systems," *IEEE Communications Magazine*, vol. 31, no. 2, pp. 28–37, February 1993.

[6] H. Shi, C. Zukowski, and O. Wing, "VLSI design optimization of input/output-buffered broadband ATM switches," in *Proc. IEEE INFOCOM*, 1996, pp. 810–817.

[7] K.J. Schultz and P.G. Gulak, "Physical performance limits for shared buffer ATM switches," *IEEE Transactions on Communications*, vol. 45, no. 8, pp. 997–1007, August 1997.

[8] H. Yamanaka, H. Saito, H. Kondoh, Y. Sasaki, H. Yamada, M. Tsuzuki, S. Nishio, H. Notani, A. Iwabu, M. Ishiwaki, S. Kohama, Y. Matsuda, and K. Oshima, "Scalable shared-buffering ATM switch with a verstile searchable queue," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 5, pp. 773–784, June 1997.

[9] A.G. Wassal and M.A. Hasan, "A VLSI switch architecture for broadband satellite networks," in *Proceedings of the IEEE Midwest Symposium on Circuits and Systems*, Notre Dame, IN, Aug. 1998, pp. 42–45.

[10] Y. Katayama, "Trends in semiconductor memories," *IEEE Micro*, vol. 17, no. 6, pp. 10–17, Nov.-Dec. 1997.

[11] C. Blondia, "A discrete-time batch Markovian arrival process as B-ISDN traffic model," *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 32, no. 3,4, pp. 3–23, 1993.

[12] S.S. Wang and J.A. Silvester, "A discrete-time performance model for integrated service ATM multiplexers," in *Proc. IEEE GLOBECOM*, 1993, pp. 757–761.

[13] J.A. Silvester, N.L.S. Fonseca, and S.S. Wang, "D-BMAP models for performance evaluation of ATM networks," in *IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, 1994, pp. 325–346.

[14] J. Roberts, U. Mocci, and J. Virtamop, Eds., *Broaband Network Teletraffic*, vol. 1155 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996, Final Report of Action COST 242.

[15] H. Michiel and K. Laevens, "Teletraffic engineering in a broadband era," *Proceedings of the IEEE*, vol. 85, pp. 2007–2033, Dec. 1997.

[16] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson, "On the self-similar nature of Ethernet-traffic," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1–15, 1994.

[17] E. Macii, M. Pedram, and F. Somenzi, "High-level power modeling, estimation, and optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 11, pp. 1061–1079, Nov. 1998.

[18] K. Muller-Glaser, K. Kirsch, and K. Neusinger, "Estimating essential design characteristics to support project planning for ASIC design management," in *Proceedings of the IEEE International Conference on Computer-Aided Design*, Los Alamitos, CA, Nov. 1991, pp. 148–151.

[19] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE Journal of Solid State Circuits*, vol. 29, no. 6, pp. 663–670, June 1994.

[20] H.B. Bakoglu, *Circuits, Interconnects, and Packaging for VLSI*, Addison Wesley, Reading, MA, 1990.

[21] S.R. Powell and P.M. Chau, "Estimating power dissipation of VLSI signal processing chips: The PFA technique," in *VLSI Signal Processing IV*, 1990, pp. 250–259.

[22] P.E. Landman and J.M. Rabey, "Activity-sensitive architectural power analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 6, pp. 571–578, June 1996.

[23] C.-T. Hsieh, C.-S. Deng, Q. Wu, and M. Pedram, "Statistical sampling and regression estimation in power macro-modeling," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, Nov. 1996, pp. 583–588.

[24] H. Mehta, R. Owens, and M.J. Irwin, "Energy characterization based on clustering," in *Proc. ACM/IEEE Design Automation Conference*, Las Vegas, NV, June 1996, pp. 702–707.

[25] A. Bogliolo, L. Benini, and G. De Micheli, "Adaptive least mean square behavioral macro modeling," in *Proc. IEEE European Design and Test Conference*, March 1997, pp. 404–410.

[26] S. Gupta and F. Najm, "Power macromodeling for high-level power estimation," in *Proc. ACM/IEEE Design Automation Conference*, Anaheim, CA, June 1997, pp. 365–370.

[27] A. Bogliolo and L. Benini, "Robust RTL power macromodels," *IEEE Transactions on Very Large Scale Integration Systems*, pp. 578–581, Dec. 1998.

[28] C.X. Huang, B. Zhang, C.D. An, and B. Swirski, "The design and implementation of POWERMILL," in *Proceedings of the ACM/IEEE International Symposium on Low-Power Design*, April 1995, pp. 105–110.

[29] P. Barth, "A Davis-Putnam based enumeration algorithm for linear pseudo-boolean optimization," Research report MPI-I-95-2-003, Max-Planck-Institut für Informatik, Germany, 1995.

[30] P. Barth and A. Bockmayr, "Modelling discrete optimisation problems in constraint logic programming," *Annals of Operations Research*, vol. 81, pp. 467–496, 1998.

[31] S.-Q. Li and H.-D. sheng, "Generalized folding algorithm for transient analysis of finite QBD processes and its queueing applications," in *Computations with Markov Chains*, W.J. Stewart, Ed., pp. 463–481. Kluwer Academic Publishers, Dordrecht, 1995.

[32] S.-Q. Li, S. Park, and D. Arifler, "SMAQ: A measurement-based tool for traffic modeling and queuing analysis: II. Network applications," *IEEE Communications Magazine*, vol. 36, no. 8, pp. 66–70, Aug. 1998.

[33] N. Akar, N.C. Oguz, and K. Shoraby, "Matrix-geometric solutions of M/G/1-type Markov chains: A unifying generalized state-space approach," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 626–639, June 1998.

[34] M.F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*, The John Hopkins University Press, Baltimore, MD, 1981.

[35] M.F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, Marcel Dekker, Inc., New York, NY, 1989.

[36] D.M. Lucantoni, "The BMAP/G/1 queue: A tutorial," in *Performance'93 and Sigmetrics'93, Lecture Notes in Computer Science – 729*, L. Donatiello and R. Nelson, Eds., pp. 330–358. Springer Verlag, 1993.

[37] J.-M. Li, I. Widjaja, and M.F. Neuts, "Congestion detection in ATM networks," *Performance Evaluation*, vol. 34, no. 3, pp. 147–168, 1998.

Amr G. Wassal (S'92-M'00) received the B.Sc. (Hons.) and M.Sc. degrees in electronics and communications engineering, both from Cairo University, Egypt, in 1993 and 1996, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo in 2000. From 1993 to 1995, he was a DSP research engineer with the Technology Group, IBM, Egypt. He is currently with the research and development division of PMC-Sierra Inc. His current research interests include modeling, analysis and design of VLSI architectures for digital communications particularly in the areas of traffic management and data security for broadband networks. Dr. Wassal is a Member of the IEEE computer and communication societies.

M.A. Hasan received the B.Sc. degree in electrical and electronic engineering, the M.Sc. degree in computer engineering, both from the Bangladesh University of Engineering and Technology, in 1986 and 1988, respectively, and the Ph.D. degree in electrical engineering from the University of Victoria in 1992.

Since 1993 he has been with the Department of Electrical and Computer Engineering, University of Waterloo, where he is now an Associate Professor. At the university of Waterloo, he is also a member of the Center for Applied Cryptographic Research and the Center for Wireless Communications. His current research interests include algorithms and architectures for computations in Galois fields, data security and reliability, and digital communication networks. From January to December of 1999, he was with the Motorola Labs., Schaumburg, IL, USA on a sabbatical leave from the University of Waterloo. Dr. Hasan is a recipient of the Raihan Memorial Gold Medal. At the University of Victoria, he was awarded the President's Research Scholarship four times. He served on the program and executive committees of several conferences, and currently, he is an associate editor of the IEEE Transactions of Computers. He is a Senior Member of the IEEE and a licensed professional engineer of Ontario.