Power Control for Crossbar-based Input-Queued Switches

Andrea Bianco, Paolo Giaccone, Guido Masera, Marco Ricca
Dipartimento di Elettronica, Politecnico di Torino, Italy

Abstract—We consider an $N \times N$ input-queued switch with a crossbar-based switching fabric implemented on a single chip. **The power consumption produced by the crossbar chip and due to the data transfer grows as NR^3, where R is the maximum bit rate.** Thus, at increasing bit rate, power dissipation is becoming more and more challenging, limiting the crossbar scalability for high performance switches.

We propose to exploit Dynamic Voltage and Frequency Scaling (DVFS) techniques to control packet transmissions through each crosspoint of the switching fabric. Our power control operates independently of the packet scheduler and exploits the knowledge of a traffic matrix obtained by on-line measurements. We propose a family of control algorithms to reduce the power consumption. The algorithms are particularly efficient in non-overloaded conditions. The actual potential of the proposed approach is also evaluated on a real design case synthesized on a 90 nm CMOS technology.

Index Terms—Input queued switch, power control, dynamic voltage frequency scaling.

1 Introduction

The aggregate bandwidth of high speed routers is growing fast, due to the increased traffic demand in the Internet. To support traffic growth, in core routers a switching fabric that must switch data at increasing speed is often implemented on a single integrated circuit. The hardware design of such fabric is becoming more and more critical, because of the large pin count and the high bit rate. Indeed, if f is the maximum digital signal frequency, the power consumption of a CMOS device is proportional to f^3 [1]. In a $N \times N$ single-chip crossbar with N^2 crosspoints, each implemented through proper logic blocks, there are[1] $\Theta(N^2)$ CMOS components (i.e., a fixed number for each crosspoint), and the total power consumption becomes proportional to R^3N, where R is the data-transmission bit rate and N is the maximum number of data simultaneously flowing across the switching fabric.

Thermal power dissipation is becoming a critical design issue, due to high integration level on a single chip, that implies very high power spatial density [2]. In integrated circuits, Dynamic Voltage and Frequency Scaling (DVFS) [1], a classical technique used to control the power consumption, is based on the idea of jointly varying the power supply voltage and the peak signal frequency. In this paper we propose to exploit DVFS for the power control of a single-chip crossbar, to reduce the power consumption at the cost of increasing packet delays at low-medium loads without sacrificing switch throughput. The main idea is to reduce the power when the traffic load is low, extending the packet transmission duration through bit voltage and frequency reduction. Indeed, networks are typically provisioned for worst-case or peak-hour traffic. However, several measurements (see for example [3]) show that backbone utilization rarely exceeds 30%, thus suggesting that exploiting low traffic conditions can be a significant asset to reduce power. We propose a set of algorithms for power control that operate on an estimated traffic matrix to assess the potential power gain that can be obtained exploiting DVFS. We take an idealized approach based on fluid model, i.e., we disregard the interaction with packet scheduling algorithms that select the packets to be transferred across the switching fabric. We only concentrate on the power of the crossbar chip, not considering the power contribution of other components of the switching architecture.

The paper is organized as follows. The system model is defined in Sec. 2, while Sec. 3 formalizes the optimal crossbar chip power control problem, describes its properties, and proposes a set of algorithms to solve it. Performance results in Sec. 4 show the possible power gain of our approach. Details on the hardware architecture for a 410 Gbps crossbar are provided in Sec. 5, where we show that the synthesis results well fit those of the theoretical model.

2 Problem Definition

We start by considering a single CMOS component, the basis of the combinatorial logic of a single crosspoint in the crossbar chip.

2.1 Energy model for a single CMOS gate

The energy consumption of a CMOS gate is strongly dependent on the supply voltage V and it can be modeled as the sum of a dynamic energy component (due to electrical signal switching activity needed to transfer

1. In Landau notation, function $g(n)$ is $\Theta(h(n))$ if, for $n \to \infty$, $k_1 h(n) < g(n) < k_2 h(n)$ for some positive constants k_1 and k_2.

sequence of 0s and 1s) and a static energy component (due to leakage currents). We consider only the dynamic energy component, while we neglect the latter contribution. Leakage currents tend to be proportional to occupied area and are normally controlled by means of circuit level techniques that are out of the scope of this work. The energy due to a bit transition (i.e., the switching activity) is a quadratic function of V according to the well known formula $E_{bit} = 0.5CV^2$, where C is the load capacitance. If we consider a 0-1 square wave signal with frequency f, the power consumption is

$$P = E_{bit} f \propto fV^2 \quad (1)$$

that represents also the thermal power to dissipate. The allowed frequency is $f \propto V$ due to the delay needed to switch from one logic state to another [4]. Thereby, the power consumption for a CMOS operating at maximum frequency and voltage is proportional to f^3. DVFS techniques jointly reduce V and f to minimize power consumption, exploiting time periods in which the signal can be "slowed down" to a lower peak frequency. This approach is actually implemented in commercial CPUs, where the processing speed changes with the instantaneous processing load [5].

We consider a CMOS device operating at voltage V, ranging between $V_{\min}$ and $V_{\max}$. Within this range, we assume that bit transmissions can occur at intermediate voltage levels. When operating at $V < V_{\max}$, since $f \propto V$, the signal frequency can be slowed down by a factor $\alpha = V_{\max}/V$ with respect to the maximum frequency allowed when using $V_{\max}$. Thus, α is the *expansion factor* of the bit duration with respect to the bit duration when using $V_{\max}$. Furthermore, V must be larger than $V_{\min} > 0$, because of technological constraints that forbid to reduce the voltage level too much and of the impact of leakage currents, that otherwise would become not negligible. Define $\beta = V_{\min}/V_{\max}$. Depending on the technology, $\beta = 0.5$ for a classical DVFS scheme or $\beta = 0.3$ in the case of an "extreme" DVFS scheme, according to [1]. By construction, $1 \leq \alpha \leq 1/\beta$.

2.2 Switching architecture

We consider in Fig. 1 an $N \times N$ input queued (IQ) switch, with virtual output queueing (VOQ), i.e. one queue VOQ_{ij} for each input i and output j pair. The IQ architecture ensures high scalability in line rate and number of ports, and the VOQ scheme is theoretically optimal from the performance point of view. The switching fabric is an $N \times N$ crossbar chip, with N^2 crosspoints and $\Theta(N^2)$ CMOS components. The crosspoint connecting input i to output j is denoted as XP_{ij} and is fed by VOQ_{ij} traffic.

A packet scheduler [6] selects the set of packets transferred simultaneously through the crossbar, satisfying the constraints that at most one packet is sent from each input and to each output, to avoid output conflicts. We do not focus on any particular scheduler, although for simplicity the model assumptions hint at packet schedulers able to achieve 100% throughput under admissible traffic. The scheduling decisions occur at a packet level, with a time granularity equal to the minimum packet duration. In the case of minimum Ethernet packet size and 10 Gbit/s line rates, a new scheduling decision must be taken every 50 ns. Given such a strict timing constraint, packet schedulers are often implemented directly in hardware, but off-chip, i.e., on a separate chip with respect to the crossbar chip.

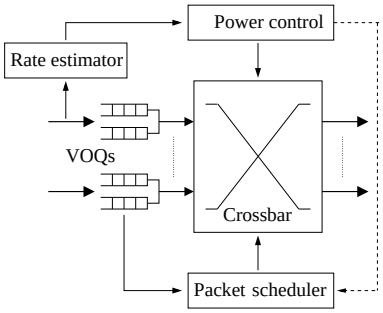

Fig. 1. Power control scheme in an IQ switch

3 Crossbar power control

The aim of the power control block in Fig. 1 is to exploit DVFS at crosspoints to reduce the crossbar chip power consumption. Based on traffic measurements on the ms scale which provide rate estimations, the control determines the DVFS factor α_{ij} for the combinatorial logic at XP_{ij}, assuming that each crosspoint is controlled independently. Due to the relaxed timing constraints, the algorithm for power control is assumed to be implemented as a software component running on an off-chip processor. Since we focus on crossbar power consumption, we disregard the power contribution of the scheduler and of the power control block. However, the only additional power consumption introduced by our proposed DVFS is due to the power control block; this contribution is negligible with respect to the scheduler consumption due to comparable algorithmic complexity and much larger time scale.

Let $\alpha = [\alpha_{ij}]$ be the $N \times N$ matrix with the DVFS factors currently employed in the crossbar. Note that setting $\alpha_{ij} > 1$ implies that the forwarding rate at XP_{ij} is reduced and the packet transmission time is increased by the expansion factor α_{ij}. This has two main consequences: i) an additional queueing delay in VOQ_{ij}, ii) the packet scheduler cannot serve any new packet from input i and to output j until XP_{ij} ends the packet transmission. Thus, the packet scheduler should be slightly modified to take into account DVFS factors in packet scheduling. We disregard this issue in the paper, and we take an ideal fluid-based approach, looking only at I/O flow rates, to evaluate the possible asymptotic benefits in terms of reduced power consumption. Note that extending packet duration might influence switch

throughput and buffer size requirements. However, the power control algorithms avoid switch overloading, by increasing packet duration only at low-medium input load. This translate in an internal load increase. In other words, the switch operates internally always in a high load regime, regardless of the real input load, but never in overload. As such, buffer requirements are not modified, because buffer size are designed for high load conditions, which are not modified by the power control scheme.

3.1 Input traffic

To avoid dealing with data content, we assume that a data packet of length L is transmitted using L signal transitions: i.e., each packet is composed by a sequence of alternating 0 and 1.

Denote the maximum line rate as $r_{\max}$, measured in [bit/s]: $r_{\max}$ is achievable only for $V = V_{\max}$. The traffic load on each link is measured on a time window whose duration T_w is in the order of ms. Let r_{ij} be the average arrival rate [bit/s] for the traffic flows enqueued at VOQ$_{ij}$ during the current time window, and $R = [r_{ij}]$ the corresponding $N \times N$ traffic matrix. Let $S = [s_{ij}]$ be the normalized traffic matrix obtained by setting $s_{ij} = r_{ij}/r_{\max}$, with $s_{ij} \in [0, 1]$. We assume that $s_{ij} > 0$ for any i and j.

Definition 1: The *average load* of matrix S is defined as

$$\rho_{\text{ave}}(S) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ij}$$

Definition 2: The *average load* at input i and at output j is $\rho_i^I(S) = \sum_{k=1}^{N} s_{ik}$ and $\rho_j^O(S) = \sum_{k=1}^{N} s_{kj}$ respectively.

Definition 3: The *maximum load* of matrix S is $\rho_{\max}(S) = \max\{\max_k\{\rho_k^I(S)\}, \max_k\{\rho_k^O(S)\}\}$.

Definition 4: The traffic matrix S is said to be *admissible* iff $\rho_{\max}(S) \leq 1$.

Obviously, $\rho_{\text{ave}}(S) \leq \rho_{\max}(S)$.

3.2 The minimum power control problem

To keep bounded queues and delays, and to avoid overload, we model the constraints related to the maximum time expansion allowed for the transmitted bits. During a measurement period, the total number of arrived bit is $T_w r_{ij}$, smaller than the maximum number of bits $T_w r_{\max}$ that can be transmitted at $V_{\max}$. Hence, the maximum allowed expansion factor for each bit is $r_{\max}/r_{ij}$, i.e. $\alpha_{ij} r_{ij} \leq r_{\max}$. At the same time, to avoid overload, it is necessary to limit the expansion at each input and output:

$$\sum_{k=1}^{N} \alpha_{ik} r_{ik} \leq r_{\max} \qquad \sum_{k=1}^{N} \alpha_{kj} r_{kj} \leq r_{\max} \qquad \forall i, j$$

which can be normalized as

$$\sum_{k=1}^{N} \alpha_{ik} s_{ik} \leq 1 \qquad \sum_{k=1}^{N} \alpha_{kj} s_{kj} \leq 1 \qquad \forall i, j \tag{2}$$

Similarly to (1), the power consumption of XP$_{ij}$, denoted as P_{ij}, is proportional to

$$P_{ij} \propto r_{ij} \left(\frac{V_{\max}}{\alpha_{ij}}\right)^2 = s_{ij} r_{\max} \left(\frac{V_{\max}}{\alpha_{ij}}\right)^2 \propto \frac{s_{ij}}{\alpha_{ij}^2}$$

The total crossbar power consumption is the sum of the power contributions of all crosspoints:

$$P_{tot} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} \propto f_P(\alpha) = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{s_{ij}}{\alpha_{ij}^2} \tag{3}$$

where $f_P(\alpha)$ is a power cost factor. Finally, the minimum power problem (denoted as OPT-MP) becomes: given an admissible S, find a feasible α minimizing f_P:

$$\min_{\alpha} f_P(\alpha) = \min_{\{\alpha_{ij} \in \mathbb{R}^+\}_{i,j}} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{s_{ij}}{\alpha_{ij}^2} \tag{4}$$

$$\text{subject to} \begin{cases} \sum_{k=1}^{N} \alpha_{ik} s_{ik} \leq 1 & \forall i \quad (5) \\ \sum_{k=1}^{N} \alpha_{kj} s_{kj} \leq 1 & \forall j \quad (6) \\ \alpha_{ij} \in \mathcal{A} & \forall i, j \quad (7) \end{cases}$$

where $\mathcal{A}$ is the set of all available voltage levels.

Property 1: OPT-MP is an integer convex non-linear optimization problem.

3.2.1 Continuous version of the problem

Following a standard methodology, we start to relax OPT-MP to continuous variables. This leads to the following problem, denoted as CONT-MP: minimize $f_P(\alpha)$ subject to (5) and (6); (7) is substituted by

$$\alpha_{ij} \geq 1 \qquad \forall i, j$$

corresponding to a DVFS scheme in which any voltage between 0 and $V_{\max}$ is allowed[2]. Let $\hat{\alpha}_{\text{OPT-MP}}$ be the optimal solution of OPT-MP. Let $\hat{\alpha}_{\text{CONT-MP}}$ be the optimal solution of CONT-MP. Since CONT-MP is a relaxed version of OPT-MP, $\hat{\alpha}_{\text{CONT-MP}}$ is a lower bound on the power cost

Property 2: $f_P(\hat{\alpha}_{\text{CONT-MP}}) \leq f_P(\hat{\alpha}_{\text{OPT-MP}})$.

Theorem 1: CONT-MP is equivalent to

$$\min_{\alpha} f_P(\alpha) \tag{8}$$

$$\text{subject to} \begin{cases} \sum_{k=1}^{N} \alpha_{ik} s_{ik} = 1 & \forall i \quad (9) \\ \sum_{k=1}^{N} \alpha_{kj} s_{kj} = 1 & \forall j \quad (10) \\ \alpha_{ij} \geq 1 & \forall i, j \quad (11) \end{cases}$$

Proof: Assume $\hat{\alpha} = [\hat{\alpha}_{ij}]$ to be the optimal solution. Define $\hat{s}_{ij} = \hat{\alpha}_{ij} s_{ij}$. By contradiction, assume that there

2. The constraint on $V_{\min}$ will be discussed at the end of the section.

exists i such that $\sum_k \hat{s}_{ik} < 1$, i.e. the i-th row of $\hat{S} = [\hat{s}_{ij}]$ sums to less than one (the same argument holds for the case the column sums to less than one). Now two cases can occur. In the first case, it exists also one column j that sums to less than one, i.e. $\sum_k \hat{s}_{kj} < 1$. Hence, it is possible to increase $\hat{s}_{ij}$ to $\hat{s}'_{ij}$ while satisfying constraints (5)-(6). The new corresponding $\alpha'_{ij} = \hat{s}'_{ij}/s_{ij}$ is feasible and provides a lower cost function; this contradicts our assumption. In the second case, all the columns sum to one and, summing over all the columns, we have $\sum_j \sum_k s_{kj} = N$, which contradicts the assumption $\sum_i \sum_k s_{ik} < N$. □

Note that one of the constraints in (9)-(10) is linearly dependent of the others and can be omitted.

Definition 5: Given a non-negative matrix $H \in \mathbb{R}^{N\times N}$, H is said to be ρ-double-stochastic if $\rho_i^I(H) = \rho_j^O(H) = \rho$ for any i and j, i.e. $\rho_{\text{ave}}(H) = \rho_{\max}(H) = \rho$. A 1-double-stochastic matrix is usually called double-stochastic matrix.

Definition 6: Given a non-negative matrix $H \in \mathbb{R}^{N\times N}$, H is said to be ρ-sub-stochastic if $\rho_{\text{ave}}(H) \leq \rho_{\max}(H) = \rho$.

Thanks to Theorem 1, CONT-MP translates to: given a ρ-sub-stochastic matrix S, find a double-stochastic matrix $\hat{S} = [\hat{s}_{ij}]$ such that the set of $\alpha_{ij} = \hat{s}_{ij}/s_{ij}$ minimizes $f_P(\alpha)$. In other words, S is augmented to become double-stochastic.

The following Theorem provides an easily computable optimal solution:

Theorem 2: Given a ρ-double-stochastic matrix S, the optimal solution $\hat{\alpha}$ for CONT-MP is $\hat{\alpha}_{ij} = 1/\rho$, for any i, j. The corresponding power cost factor is $f_P(\hat{\alpha}_{\text{CONT-MP}}) = N\rho^3$.

Proof: The proof is based on the use of the Lagrange multipliers and on the Taylor's Theorem for multivariate functions. Denote $\otimes$ as the Hadamard product (i.e., element-by-element) of two matrices. Define $\hat{\alpha}$ as the optimal solution given by $\hat{\alpha}_{ij} = 1/\rho$ and define α, with $\alpha \neq \hat{\alpha}$, a generic feasible solution satisfying (9) and (10); $\alpha \otimes S$ and $\hat{\alpha} \otimes S$ are both double stochastic matrices. We can define matrix $\Delta = \alpha - \hat{\alpha}$ and assume that $\max_{i,j}\{\Delta_{ij}\} \leq \epsilon$ where $\epsilon > 0$. We can use Birkhoff-von Neumann Theorem [7] to claim that there exist a set of real numbers γ_k such that

$$\Delta \otimes S = \sum_k \gamma_k M^k \qquad \sum_k \gamma_k = 0 \tag{12}$$

where M^k is a permutation matrix. Equivalently,

$$\Delta_{ij} = \sum_k \gamma_k \frac{m_{ij}^k}{s_{ij}} \tag{13}$$

Consider for algebraic convenience consider the vectorization form of a matrix; the column vector form of matrix Δ is denoted by $\underline{\Delta}$. By classical Taylor's Theorem for multivariate functions,

$$f_P(\alpha) - f_P(\hat{\alpha}) = \underline{\Delta}^T \nabla f_P(\hat{\alpha}) + \frac{1}{2}\underline{\Delta}^T H(\underline{\eta})\underline{\Delta} \tag{14}$$

where $H(\underline{\eta})$ is the Hessian matrix computed in $\eta = (1-\gamma)\hat{\alpha} + \gamma\alpha = \hat{\alpha} + \gamma\Delta$, for some constant $\gamma \in [0,1]$. Equivalently,

$$\eta_{ij} = \hat{\alpha}_{ij} + \gamma\Delta_{ij} \tag{15}$$

We first show that the first term in the right hand side of (14) is null. Indeed, by (12) and (13):

$$\underline{\Delta}^T \nabla f_P(\hat{\alpha}) = \sum_{ij} \frac{-2s_{ij}}{\hat{\alpha}_{ij}^3} \sum_k \gamma_k \frac{m_{ij}^k}{s_{ij}} = \sum_{ij}(-2\rho^3) \sum_k \gamma_k m_{ij}^k = (-2\rho^3) \sum_k \gamma_k \sum_{i,j} m_{ij}^k = (-2\rho^3) \sum_k \gamma_k N = 0$$

Let us consider now the second term in the right hand side of (14). Observe that $H(\underline{\alpha})$ is a diagonal matrix, in which the element corresponding to (i,j) pair is equal to $6s_{ij}/\alpha_{ij}^4$. Hence, by (15):

$$\underline{\Delta}^T H(\underline{\eta})\underline{\Delta} = \sum_{i,j} \Delta_{ij}^2 \frac{6s_{ij}}{\eta_{ij}^4} = \sum_{i,j} \Delta_{ij}^2 \frac{6s_{ij}\rho^4}{(1+\gamma\rho\Delta_{ij})^4}$$

Let $\epsilon' = \min_{ij}\{\Delta_{ij} | \Delta_{ij} > 0\}$ and $s' = \min_{ij}\{s_{ij}\}$. Finally, we can claim

$$f_P(\alpha) - f_P(\hat{\alpha}) = \underline{\Delta}^T H(\underline{\eta})\underline{\Delta} \geq \frac{6\rho^4(\epsilon')^2 s'}{(1+\gamma\rho\epsilon)^4} > 0$$

that means that any $\alpha \neq \hat{\alpha}$ that satisfies (9) and (10) cannot be the optimal solution.

The minimum power cost factor is immediately obtained by computing $f_P(\hat{\alpha})$. □

In Sec. 5, we validate the cubic relation between power and load through the results of the actual hardware synthesis of a crossbar chip. Furthermore, we can get an important intuition from the above theorem, which will drive the design of approximated algorithms for the CONT-MP problem: *In the optimal solution, all the α_{ij} are expanded proportionally by the same factor.*

When considering also the constraint on $V_{\min}$, the expansion ratio is limited by $\alpha_{ij} \leq 1/\beta$. For ρ-double-stochastic matrices, the optimal solution becomes $\alpha_{ij} = \min(1/\rho, 1/\beta)$, $\forall i, j$ and the corresponding optimal solution for CONT-MP becomes:

$$f_P(\hat{\alpha}_{\text{CONT-MP}}) = \begin{cases} N\rho\beta^2 & \text{if } \rho < \beta \\ N\rho^3 & \text{if } \rho \geq \beta \end{cases} \tag{16}$$

Thus, β is the value of "critical load" above which DVFS is not able to expand the bit duration due to the constraints imposed by the traffic load in (2).

Consider now a relaxed version of the CONT-MP problem, denoted as MISO-MP (Multiple-Inputs Single-Output), in which we remove the expansion constraints (9) on each input.

Theorem 3: Given any admissible traffic matrix S, the optimal solution of MISO-MP is given by $\alpha_{ij} = 1/\rho_j^O(S)$. The corresponding power cost factor is:

$$f_P(\hat{\alpha}_{\text{MISO-MP}}) = \sum_j (\rho_j^O(S))^3$$

Note that this results does not require S to be a double-stochastic matrix.

Proof: Define the Lagrange function as

$$\Lambda = \sum_{ij} s_{ij}/\alpha_{ij}^2 + \sum_j \lambda_j \big(\sum_k s_{kj}\alpha_{kj} - 1 \big)$$

A necessary condition for the solution to be optimal is $\partial\Lambda/\partial\alpha_{ij} = 0$, which implies $-2s_{ij}\alpha_{ij}^{-3} + \lambda_j \alpha_{ij} s_{ij} = 0$. It should be $\alpha_{ij} = (2/\lambda_j)^{-4}$, i.e. for a fixed j, all the α_{ij} are constant. Thus (10) becomes $\alpha_{ij} \sum_k s_{kj} = 1$ and hence $\alpha_{ij} = 1/\rho_j^O(S)$. This satisfies also (11). By simple substitution, we get the corresponding power cost. □

Property 3: $f_P(\hat{\alpha}_{\text{MISO-MP}}) \leq f_P(\hat{\alpha}_{\text{CONT-MP}})$
i.e. MISO-MP provides a lower bound, simple to compute, for CONT-MP and OPT-MP under any admissible traffic matrix.

3.2.2 Power consumption without DVFS

A feasible, but not optimal, solution for OPT-MP is when no DVFS scheme is adopted, i.e. $\alpha_{ij} = 1$ for all i, j. We define this scheme as NODVFS and the corresponding solution as $\hat{\alpha}_{\text{NODVFS}}$. The power cost factor f_P under any admissible traffic matrix S can be obtained by setting $\alpha_{ij} = 1$ in (3):

$$f_P(\hat{\alpha}_{\text{NODVFS}}) = \sum_{i=1}^{N}\sum_{j=1}^{N} s_{ij} = N\rho_{\text{ave}}(S) \qquad (17)$$

denoting a linear relationship between the average load on S and the total power consumption.

Property 4: $f_P(\hat{\alpha}_{\text{OPT-MP}}) \leq f_P(\hat{\alpha}_{\text{NODVFS}})$.
Thus $f_P(\hat{\alpha}_{\text{NODVFS}})$ is a loose upper bound for OPT-MP.

We define the *relative power* $\eta(\hat{\alpha})$ of a DVFS solution $\hat{\alpha}$, relative to NODVFS, as:

$$\eta(\hat{\alpha}) = \frac{f_P(\hat{\alpha})}{f_P(\hat{\alpha}_{\text{NODVFS}})} = \frac{f_P(\hat{\alpha})}{N\rho_{\text{ave}}(S)}. \qquad (18)$$

Since $\eta(\hat{\alpha}) \in [0, 1]$, the closer $\eta(\hat{\alpha})$ to zero, the larger the scheme gain with respect to NODVFS.

For double-stochastic matrices, dividing (17) by (16):

Property 5: Under ρ-double-stochastic matrices, $\eta(\hat{\alpha}_{\text{CONT-MP}}) = \beta^2$ for $\rho < \beta$, ρ^2 for $\rho \geq \beta$.

In summary, the solution to the CONT-MP problem, which uses any voltage level between $V_{\min}$ and $V_{\max}$, provides a lower bound for the power of the OPT-MP problem. When the matrix is double-stochastic, the optimal solution to CONT-MP is trivial. Otherwise, a lower bound can be found with the solution of MISO-MP, trivial to compute.

3.3 Power control algorithms

To solve OPT-MP for any traffic matrix we propose to: i) solve the corresponding CONT-MP problem, ii) approximate each α_{ij} to the closest smaller value available in the set $\mathcal{A}$. In other words, if α_{ij} is the solution for CONT-MP, then use for OPT-MP:

$$\alpha'_{ij} = \max\{\alpha \in \mathcal{A} \mid \alpha \leq \alpha_{ij}\}$$

Note that, by construction, the set of α'_{ij} defines an admissible solution for OPT-MP.

To solve CONT-MP, we adopt a quasi-optimal algorithm based on the logarithmic barrier method for convex problems [8] which provides an ϵ-approximation of the optimal solution. Furthermore, we adopt a two-steps algorithm: we augment S to a double stochastic $\hat{S}$ according to one of algorithms among AUGM-1, AUGM-MAX or AUGM-SORT, described below. Then, we compute $\alpha_{ij} = \hat{s}_{ij}/s_{ij}$.

INCREASE-MATRIX Algorithm
Input: $N \times N$ matrix $S = [s_{ij}]$, $\{\rho_i^I\}_{i=1}^N$, $\{\rho_j^O\}_{j=1}^N$, ρ_T, Ω^I, Ω^O.
Output: $N \times N$ matrix $\Delta = [\delta_{ij}]$
1. $\delta_{ij} = 0$ for any $1 \leq i, j \leq N$
2. $\Omega^{IO} = \{(i, j) : i \in \Omega^I, j \in \Omega^O\}$
3. **repeat** until no choice is anymore available
4. **choose** any $(i, j) \in \Omega^{IO}$ such $\max\{\rho_i^I, \rho_j^O\} < \rho_T$
5. $\delta_{ij} = \min\{\rho_T - \rho_i^I, \rho_T - \rho_j^O\}$
6. $\rho_i^I = \rho_i^I + \delta_{ij}, \quad \rho_j^O = \rho_j^O + \delta_{ij}$

We now describe the INCREASE-MATRIX procedure, on which all the augmentation algorithms are based. The inputs of the procedure are i) a sub-stochastic matrix S, ii) the corresponding row ρ_i^I and column ρ_j^O sums; iii) a target load value ρ_T such that $\max_k\{\rho_k^I, \rho_k^O\} \leq \rho \leq 1$, and iv) a set of input ports Ω^I and output ports Ω^O. The algorithm returns a matrix $\Delta = [\delta_{ij}]$ with the largest possible elements such that: (i) only the elements δ_{ij} corresponding to rows and columns present in both Ω^I and Ω^O may be > 0; (ii) the maximum row and column sum is ρ_T, i.e.

$$\sum_{k=1}^{N} s_{ik} + \delta_{ik} \leq \rho_T \text{ for any } i \in \Omega^I$$
$$\sum_{k=1}^{N} s_{kj} + \delta_{kj} \leq \rho_T \text{ for any } j \in \Omega^O$$

The algorithm operates only on a sub-matrix restricted to the rows in Ω^I and the columns in Ω^O. It chooses a sequence of elements whose row and column sum to less than ρ_T. Then, each element in the sub-matrix is augmented to reach ρ_T without violating the constraints. Note that the maximum number of iterations in step 3 is upper bounded by $2N$.

Having defined INCREASE-MATRIX, we now describe the algorithms we propose to augment S to a double-stochastic $\hat{S}$:

- AUGM-1:
  1) compute ρ_i^I and ρ_j^O for any i and j;
  2) run INCREASE-MATRIX on S, ρ_i^I, ρ_j^O, $\rho_T = 1$, $\Omega^I = \Omega^O = \{1, \ldots, N\}$;
  3) compute $\hat{s}_{ij} = s_{ij} + \delta_{ij}$ for all i and j.

Note that AUGM-1 is a classical iterative algorithm (see Sec. II.A of [7]) to augment a sub-stochastic matrix to a double-stochastic one. The complexity is $O(N^2)$, due to steps 1) and 3).

- AUGM-MAX:
  1) compute ρ_i^I and ρ_j^O for any i and j;

2) compute $\rho_{\max}(S)$;
3) run INCREASE-MATRIX on S, ρ_i^I, ρ_j^O, $\rho_T = \rho_{\max}(S)$, $\Omega^I = \Omega^O = \{1, \ldots, N\}$;
4) compute $\hat{s}_{ij} = s_{ij} + \delta_{ij} + (1 - \rho_{\max}(S))/N$.

The complexity of AUGM-MAX is $O(N^2)$, due to steps 1) and 4).

- AUGM-SORT:

1) compute ρ_i^I and ρ_j^O on S for any i and j;
2) sort ρ_i^I and ρ_j^O in increasing order. Let $i_{(k)}$ be the kth input and $j_{(k)}$ be the kth output in such increasing sequences;
3) initialize an auxiliary matrix $X^{(0)} = S$ and set $\Omega_0^I = \Omega_0^O = \emptyset$;
4) iterate, for k from 1 to N, the following steps:
   a) $\Omega_k^I = \Omega_{k-1}^I \cup i_{(k)}$, i.e. the set of the inputs with the k smallest row sums;
   b) $\Omega_k^O = \Omega_{k-1}^O \cup j_{(k)}$, i.e. the set of the outputs with the k smallest column sums;
   c) run INCREASE-MATRIX on $X^{(k-1)}$, ρ_i^I, ρ_j^O, Ω_k^I, Ω_k^O and $\rho_T^{(k)} = \max\{\rho_{i_{(k)}}^I, \rho_{j_{(k)}}^O\}$, i.e. $\rho_T^{(k)}$ is the maximum load for the first kth inputs and outputs of S;
   d) $x_{ij}^{(k)} = x_{ij}^{(k-1)} + \delta_{ij}$ for any i, j, i.e. set $X^{(k)} = X^{(k-1)} + \Delta$;
   e) eventually go to a) to start a new iteration;
5) compute $\hat{s}_{ij} = x_{ij}^{(N)} + (1 - \rho_{\max}(X^{(N)}))/N$.

The complexity of AUGM-SORT is $O(N^2)$ by optimizing the data structure to choose an $(i, j) \in \Omega^{IO}$ in INCREASE-MATRIX and by sorting only once ρ_i^I and ρ_j^O.

Theorem 2 suggests that the optimal way to increase the S is proportionally, at least for some families of traffic. AUGM-1 is a classical way to augment a matrix. Instead, AUGM-MAX and AUGM-SORT tend to augment the matrix more proportionally.

4 PERFORMANCE EVALUATION

We first discuss the performance for ρ-double-stochastic matrices. Then, we move to ρ-sub-stochastic matrices.

4.1 Power consumption for double-stochastic matrices

According to Theorem 2, the optimal solution for CONT-MP is expressed by (16). Fig. 2 shows the *power consumption per port* $f_P(\hat{\alpha})/N$ vs. the average load, for the optimal solution of CONT-MP and $\beta \in \{0.3, 0.5, 0.7\}$. We show also the linear growth of NODVFS, computed with (17). For small loads, DVFS is very efficient, by reducing the power by a factor $1/\beta^2$ (see Property 5), equal to 11, 4 and 2, respectively, for each value of β. For larger loads, the DVFS power reduction decreases, becoming negligible in highly loaded conditions, because bit expansion is not allowed due to the high traffic load.

We now consider the effect of a finite set $\mathcal{A}$ of voltage levels. Table 1 shows the worst-case (for any load) ratio between the consumption of OPT-MP with finite set of voltage levels and the consumption of CONT-OPT with continuous DVFS, as a function of the number of available voltage levels. The $|\mathcal{A}| - 2$ intermediate voltage levels between $V_{\min}$ and $V_{\max}$ have been numerically optimized to minimize such ratio. Note that very few intermediate levels (i.e., one for $\beta = 0.5$) are sufficient to observe differences lower than 10%. Hence, the simple solution to CONT-MP well approximates the solution to the OPT-MP problem. Finally, very few voltage levels are enough to exploit the potential benefits of DVFS.

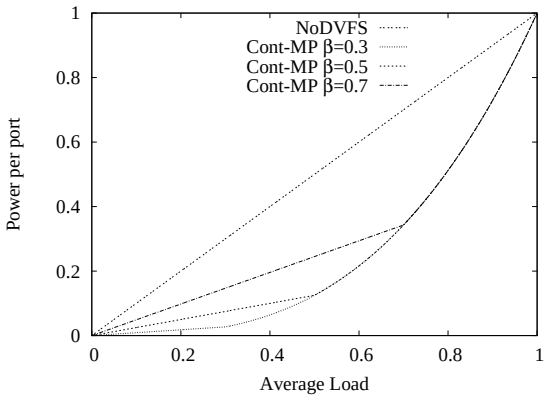

Fig. 2. Optimal solution for continuous DVFS (CONT-MP), under any ρ-double-stochastic matrix.

TABLE 1
The power consumption ratio between DVFS with discrete voltage levels (OPT-MP) and continous DVFS (CONT-MP), for double-stochastic matrices

$\lvert\mathcal{A}\rvert$	β	Voltage levels $/V_{\max}$	$\max_{0 \le \rho \le 1} \frac{f_P(\hat{\alpha}_{\text{OPT-MP}})}{f_P(\hat{\alpha}_{\text{CONT-MP}})}$
3	0.3	$0.3, 0.55, 1$	1.31
	0.5	$0.5, 0.71, 1$	1.09
	0.7	$0.7, 0.84, 1$	1.02
4	0.3	$0.3, 0.45, 0.67, 1$	1.13
	0.5	$0.5, 0.63, 0.79, 1$	1.04
	0.7	$0.7, 0.78, 0.89, 1$	1.01
5	0.3	$0.3, 0.41, 0.55, 0.74, 1$	1.07
	0.5	$0.5, 0.60, 0.71, 0.84, 1$	1.02
	0.7	$0.7, 0.76, 0.84, 0.92, 1$	1.01

4.2 Power consumption for sub-stochastic matrices

We consider the family of random traffic matrices generated as follows. Given $\rho \in (0, 1]$, generate a matrix $U = [u_{ij}]$ of N^2 random variables, uniformly distributed on the interval $(0, 1]$. Then, derive each element of S as $s_{ij} = u_{ij}\rho/(\rho_{\max}(U))$. Using this construction, it can be shown that the corresponding average load $\rho_{\text{ave}}(S) \approx \rho/(1 + (\sqrt{0.67 \log(N)/N}))$ for large enough N.

We compare the algorithms proposed in Sect. 3.3 for continous DVFS, because, as shown in the previous section, CONT-MP is a good approximation of OPT-MP even when few voltage levels are available. We show the optimal solution for CONT-MP only for smaller switch sizes ($N = 16$), due to computational constraints. We report also the solution for the lower bound provided by

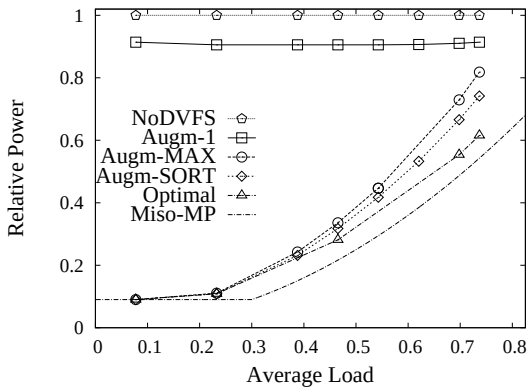

Fig. 3. Relative power for $N = 16$ and $\beta = 0.3$, under sub-stochastic matrices

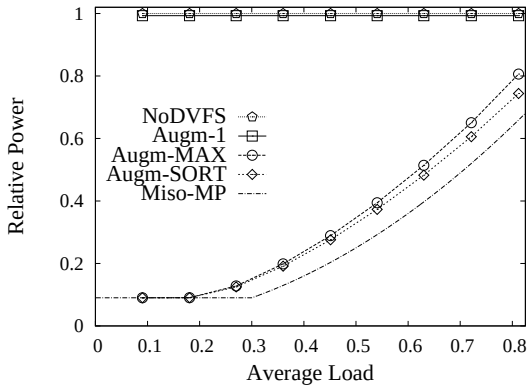

Fig. 4. Relative power for $N = 256$ and $\beta = 0.3$, under sub-stochastic matrices.

MISO-MP. Even if the results hold for $\beta = 0.3$, similar results were obtained for other values of β.

Figs. 3, 4 show the relative power (Eq. (18)), for different N. Note that, to ensure admissibility, the maximum average load in the abscissa is limited by construction to be always less than $1/(1+\sqrt{0.67\log(N)/N})$, i.e. 0.75 and 0.88 for $N = 16$ and $N = 256$ respectively.

When increasing $\rho_{\text{ave}}(S)$, the relative power of MISO-MP shows a quadratic growth, similarly to double-stochastic matrices for which Property 5 holds. The behavior is close to the optimal solution, justifying its use to approximate CONT-MP for large N. Even if not optimal, AUGM-SORT and AUGM-MAX show performance close to the optimal. On the contrary, AUGM-1 behaves the worst, only providing minor power reductions with respect to NODVFS.

Similar results holds For $N = 256$ in Fig. 4. We were unable to obtain the optimal solution in reasonable time. AUGM-1 does not provide any benefit. AUGM-SORT and AUGM-MAX provide performance close to the lower bound MISO-MP. Thus, these DVFS schemes appear to be the most efficient, especially at low average load, regardless of the switch size.

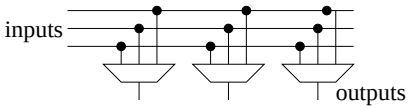

Fig. 5. Mux-based 3×3 crossbar

5 HARDWARE DESIGN AND EVALUATION

To better explore the effects of DVFS on a real switch fabric, a 128×128 crossbar switch was adopted as a case study. To optimize crossbar scalability, instead of the classical X-Y architecture, we choose a mux-tree based pipelined architecture. Indeed, in classical X-Y based crossbar switches [9], any input–output connection is provided by horizontal and vertical wires spanning the whole area. Hence, propagation delay along wires tends to grow rapidly with the number of input-output ports and soon becomes the limiting factor for throughput performance. Multiple bit slices can be used to cope with limited clock frequency, while reaching at the same time high line throughput. However, in this case, improved performance comes at the cost of additional implementation complexity.

High data rates over a large switch, with more than one hundred input output ports, can be obtained at a lower implementation complexity with a mux-tree based pipelined architecture [9], shown in Fig. 5: Each output is connected through a *tree* of multiplexers that receive all input ports. Two basic features of the tree organization can be exploited to improve speed: (i) the entire multiplexing operation can be split in several tree stages, with each stage individually sized to match timing constants according to its load capacitance, and (ii) pipeline registers can be inserted along the tree to cut critical path delays, thus achieving very high clock frequency.

The mux–tree based pipelined switch of size 128×128 was modeled using VHDL language and synthesized to derive area occupation, achievable throughput and dissipated power. Fig. 6 shows the structure of a single slice of the crossbar fabric: each input port receives data serially and the 128 inputs are divided into two parts, where the upper (and the lower) portion deals with 64 inputs. Internal registers are used to provide pipelining. In the upper half of the fabric, 16 multiplexers and 4 multiplexers are contained in the first and second pipeline stages respectively. A 4×1 multiplexer is allocated in the third pipeline stage. The same structure is repeated in the lower half, and a 2×1 multiplexer is used for the final selection. Thus, the showed slice forms a 128×1 multiplexer with pipelining. To control the whole set of multiplexers, 85 select lines are required.

The complete fabric architecture consists of 128 slices equal to the one given in Fig.6. The same data inputs are applied to each slice and a total of $128 \times 85 = 10880$ select lines are used to control the switch. Destination conflicts are not allowed in the described architecture, and are prevented by a proper scheduling algorithm [6].

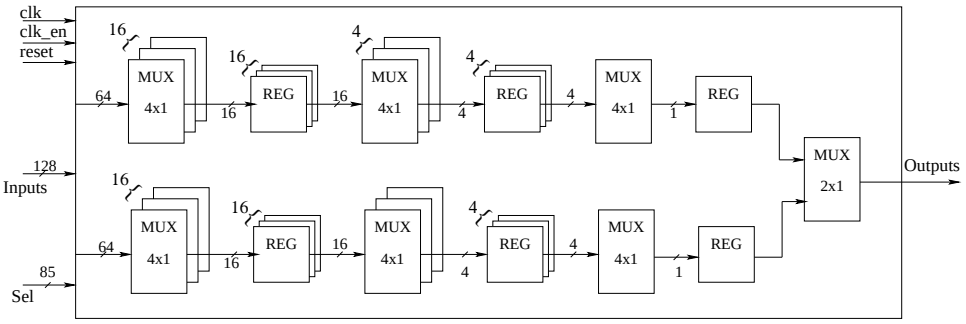

Fig. 6. Architecture of a slice of the switch fabric

A further important property of the adopted switch fabric architecture is its modularity. This feature enables the possibility to adopt a hierarchical synthesis flow that simplifies the floorplan. Additionally, although this is not exploited in this work, the modular structure of the switch also allows for applying different choices of voltage and frequency scaling to individual slices. Assuming that a lower traffic is observed along paths associated with a specific slice, then voltage and frequency scaling for this single slice would be beneficial to reduce power consumption and would allow at the same time for higher throughput across different slices.

The VHDL code of the fabric was written, debugged and simulated under Mentor Graphics Modelsim using randomly generated patterns of input data. Synthesis was performed using Synopsys Design Compiler on a 90 nm CMOS technology. The power analysis of the switch fabric was performed using Synopsys Power Compiler. We do not consider the power contribution due to the implementation of the power control algorithm or any other component because we focus on the crossbar chip. We restrict our analysis to the synthesis results and we do not consider the consumption due to the actual chip layout; hence, our power consumption results are relative. Derived power dissipation figures are based on the actual switching activities measured at circuit nodes during simulation of the fabric in the presence of different test patterns. Thanks to the high level of applied pipelining, the maximum operating frequency of the designed crossbar, when the supply voltage is not scaled, is as high as 3.2 GHz, allowing to reach an aggregated bandwidth of 410 Gbps. To evaluate the potential of the described DVFS approach, the crossbar was synthesized with several values of supply voltage and frequency of the clock signal. Six scaling factors (i.e. $\{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, corresponding to $\alpha = \{2.50, 2.00, 1.67, 1.43, 1.25, 1.11\}$), were used to reduce supply voltage. In addition, the clock frequency, f_{CK}, was changed in the range between the maximum achievable value of 3.2 GHz down to 200 MHz, equally for all the ports. Hence, the corresponding traffic matrix S is ρ-double-stochastic with all $s_{ij} = \rho/N$ and $\rho = f_{CK}/(3.2\text{ GHz})$. The power consumption in the fabric is associated with the switching activity in the slice components and therefore to the

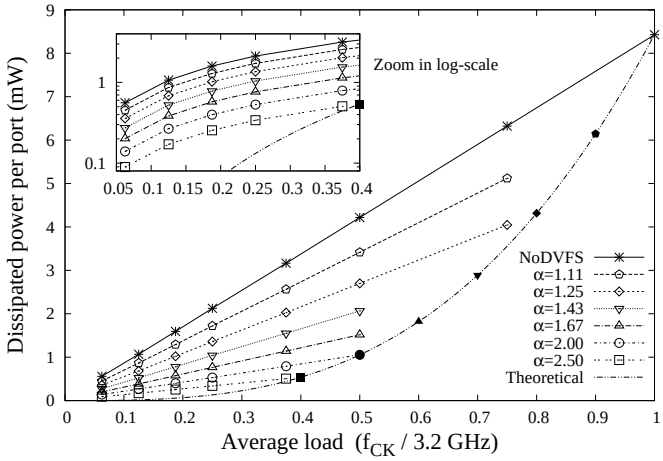

Fig. 7. Power obtained by the VHDL synthesis, for a 128×128 crossbar with 410 Gbps bandwidth.

average data throughput. For each selected value of f_{CK}, the maximum possible data rate has been assumed for input data. For example, with $f_{CK} = 1.2$ GHz, data are received at the rate of 1,200 Mbps per input port. The select lines which control the multiplexers are assumed to switch at a 1000 times lower rate. Note that power would also be consumed to change between voltage levels. Furthermore, each transition to new values of supply voltage and f_{CK} introduces a latency, which may affect the global throughput. However, for simplicity reasons, latency and power overheads generated by these transitions are not considered in this study.

Switch fabric power consumption per port is reported in Fig. 7 for different voltage scaling factors and clock frequencies. The theoretical curve is $\rho_{\text{ave}}(S)^3$. As expected, power consumption scales linearly with f_{CK} and thus with input data rate, but the slope depends on the applied voltage scaling. Therefore different power reduction gains can be obtained at different input data rates. For example, if input data rate is 50% of the maximum allowed level, 75% of the dissipated power can be saved, from 4.2 mW with no applied DVFS to 1 mW with a voltage scaling factor equal to 0.5. A lower reduction of dissipated power is possible when working at higher data rates: with input data at 75% of

the maximum frequency, the dissipated power can be reduced by 51% from 6.3 mW to 3.1 mW.

Furthermore, the filled points on the theoretical curve for a specific load ρ are aligned with the linear interpolation of the powers obtained for a specific value of $\alpha = 1/\rho$. This means that the cubic dissipation model of Theorem 2, based on a single expansion factor for the whole crossbar, is accurate.

6 Conclusions

We discussed the potential power gains that DVFS techniques can provide when controlling a crossbar used as a switching fabric in an input-queued switch. We took an idealized approach, disregarding the details related to packet scheduling, looking at flow rates.

Performance results, validated through a real hardware synthesis, show that a significant power reduction can be obtained, especially at low loads. The proposed algorithms are computationally simple and obtain performance gain close to those of more complex, optimal algorithms.

References

[1] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Trans. on VLSI Systems*, vol. 13, no. 11, pp. 1239–1252, Nov. 2005.

[2] F. Hameed, M. Faruque, and J. Henkel, "Dynamic thermal management in 3d multi-core architecture through run-time adaptation," in *IEEE Design, Automation & Test in Europe (DATE)*, 2011.

[3] https://research.sprintlabs.com/packstat/packetoverview.php.

[4] M. Flynn and P. Hung, "Microprocessor design issues: thoughts on the road ahead," *IEEE Micro*, vol. 25, no. 3, pp. 16–31, May 2005.

[5] T. Kolpe, A. Zhai, and S. Sapatnekar, "Enabling improved power management in multicore processors through clustered dvfs," in *IEEE Design, Automation & Test in Europe (DATE)*, 2011.

[6] H. J. Chao and B. Liu, *High Performance Switches and Routers*. Wiley-IEEE Press, 2007.

[7] C.-S. Chang, W.-J. Chen, and H.-Y. Huang, "Birkhoff-von neumann input buffered crossbar switches," in *IEEE INFOCOM*, vol. 3, March 2000, pp. 1614–1623.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[9] Ting Wu, Chi-Ying Tsui, and Mounir Hamdi, "A 2Gb/s 256 x 256 CMOS crossbar switch fabric core design using pipelined MUX," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Phoenix-Scottsdale, AZ, May 2002.