

# Variation-Adaptive Feedback Control for Networks-on-Chip with Multiple Clock Domains

Umit Y. Ogras, Radu Marculescu, Diana Marculescu

Department of Electrical and Computer Engineering  
Carnegie Mellon University, Pittsburgh, PA, USA

e-mail: {uogras,radum,dianam}@ece.cmu.edu

## ABSTRACT

This paper discusses the use of networks-on-chip (NoCs) consisting of multiple voltage-frequency islands to cope with power consumption, clock distribution and parameter variation problems in future multiprocessor systems-on-chip (MPSoCs). In this architecture, communication within each island is synchronous, while communication across different islands is achieved via mixed-clock, mixed-voltage queues. In order to dynamically control the speed of each domain in the presence of parameter and workload variations, we propose a robust feedback control methodology. Towards this end, we first develop a state-space model based on the utilization of the inter-domain queues. Then, we identify the theoretical conditions under which the network is controllable. Finally, we synthesize state feedback controllers to cope with workload variations and minimize power consumption. Experimental results demonstrate robustness to parameter variations and more than 40% energy savings by exploiting workload variations through dynamic voltage-frequency scaling (DVFS) for a hardware MPEG-2 encoder design.

## Categories and Subject Descriptors

B.7 [Hardware]: Integrated circuits.

## General Terms

Algorithms, Design.

## Keywords

Voltage-frequency island, dynamic voltage-frequency scaling, parameter variation, MPSoC, networks-on-chip, feedback control.

## 1. INTRODUCTION

Due to continuous scaling of CMOS technology, systems containing a large number of processors and on-chip memories have become a reality [14]. Consequently, novel architectures that allow various cores communicate to each other via the Network-on-Chip (NoC) paradigm have emerged as a promising alternative to traditional bus-based approaches [3]. By eliminating global wires, the NoC approach provides the needed scalability and predictability, while facilitating design reuse.

In addition to increasing complexity, systems designed in nanoscale technologies suffer from systematic and random variations in process, voltage, and temperature (PVT). While the threshold voltage variations result in chips with widely varying leakage and fre-

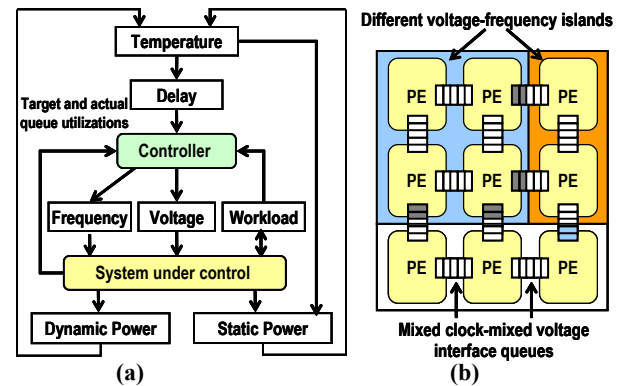


Figure 1 (a) Loop showing how the variations in temperature and voltage affect the system power consumption which eventually feeds back to temperature. (b) An NoC architecture with multiple voltage-clock domains are shown.

quency values, the dynamic variations in the workload and temperature cause uneven voltage distribution and hence, performance mismatches in the system [1]. This in turn results in variations in power consumption and affects the chip temperature generating a feedback loop as shown in Figure 1(a). Adaptive body biasing and frequency binning are used to cope with leakage and frequency variations across different chips [7]. However, *within die* variations play an increasingly important role in system power consumption and performance as the technology scales down [1]. Since designers cannot rely on the accuracy of the nominal parameter values, there is a tremendous need for *on-line* techniques that can cope with such dynamic variations. More precisely, there is a need for efficient algorithms and built-in circuitry able to adapt the system behavior to workload variations (see Figure 1(a)) and, at the same time, cope with the parameter variations which cannot be predicted or accurately modeled at design time.

Towards this end, we explore distributed, variation-adaptive control techniques that can improve the performance, power consumption, and reliability of future NoC architectures in a synergistic manner. We consider NoC architectures consisting of multiple voltage-frequency islands (VFIs), or voltage-clock domains, as shown in Figure 1(b). Each island is a synchronous region, *i.e.*, the cores within the same island share a single clock and supply voltage that can be controlled independently of other islands. Communication across different clock domains is asynchronous and it can be achieved via mixed-clock mixed-voltage FIFOs [2]. We use the utilization of these queues as a performance metric to adapt the voltage and frequency values of various islands.

The contribution of our work is twofold: First, we develop a *state-space model* for multiple voltage-clock domain NoC designs, and identify the *theoretical conditions* under which such a system is controllable and stable. Then, we propose a novel methodology

to *dynamically control* the speed of each island, as shown in Figure 1(a). The controller not only considers the dynamic workload variations, but also ensures that the operating frequency does not exceed the maximum allowed value for a given temperature. Within this framework, we explore two feedback control techniques, namely regulation and tracking, as follows:

1) *On-line regulation*: For this scenario, we assume that the *nominal* values of the operating voltage and frequency are determined by an algorithm running at a high level of abstraction (e.g., [10]). The *actual* operating conditions such as voltage, frequency and temperature are usually different than the values assumed at design time due to PVT and dynamic workload variations [1]. Hence, optimizations carried out for nominal values are likely to result in poor performance figures. Our goal is to regulate the voltage and clock frequency *around* the nominal values and counteract the effects of parameter variation and uncertainties that are too expensive or too difficult to be characterized and dealt with at design time. Regulating the voltage and clock frequency around nominal values makes sense also from a technology perspective, since the window of change for the supply voltage is continuously shrinking [14]. Finally, this approach enables the re-use of existing off-line Dynamic Voltage and Frequency Scaling (DVFS) algorithms that will otherwise fail due to the PVT variations.

2) *On-line tracking*: This scenario is more general, in the sense that it can scale the voltage and clock frequency within a *wider range* of values in order to minimize the power consumption, as well as provide robustness in operation. In this case, we assume there exists a lower bound on the operating frequency (hence voltage) for a subset of voltage-frequency islands that satisfies certain deadline or rate constraints. The proposed controller sets the speed of each island such that the constraints on the minimum operating values are satisfied. For instance, an audio/video decoder has to maintain a minimum level of decoding rate such that the output can be displayed without distortion. In this case, the proposed methodology takes this minimum output rate as input. The voltage-frequency island that generates the output stream works at the minimum permissible speed, while the speeds of all other islands are adjusted dynamically to match the desired output rate. As detailed in Section 4, the proposed controller throttles the clock frequency to exploit the dynamic variations in the workload, thereby saving about 46% of power without any loss in performance.

The remainder of this paper is organized as follows. In Section 2, we review the related work and highlight our contributions. Section 3 overviews the proposed framework and provides a detailed analysis. Experimental results are included in Section 4. Finally, Section 5 concludes the paper.

## 2. RELATED WORK

Generally speaking, feedback control reduces the systems sensitivity to parameter variations. So, design of nanoscale MPSoCs is likely to benefit from the systematic approaches from modern control theory [5,11]. As such, in [9], the authors propose using adaptive body biasing and dynamic voltage scaling simultaneously to reduce power in processors with a single clock domain. Partitioning NoCs into multiple voltage-clock islands to minimize the energy consumption is considered in [10]. Power management for NoCs have been recently considered by several authors. The work in [13] presents a stochastic power management technique for NoCs. In [12], the authors present a run-time technique to satisfy

**Table 1: Notation used in the paper**

Symbol	Explanation
$q_i(t)$	Occupancy of queue $i$ at time $t$
$Q(t)$	State vector containing the occupancy of all queues.
$\lambda_i(k)$ ( $\mu_i(k)$ )	Average arrival (departure) rate to (from) queue $i$ in the $k^{\text{th}}$ control interval
$f_j(k)$ ( $V_j(k)$ )	Frequency (voltage) of domain $j$ in the $k^{\text{th}}$ control interval
$\bar{\lambda}_i$ , ( $\bar{\mu}_i$ )	Average write (read) operations per cycle for queue $i$ , e.g. $\lambda_i(k) = \bar{\lambda}_i f_j(k)$ , where the input of queue $i$ is in island $j$ .
$K$ , $K_0$ , $K_1$	The gain matrices for the feedback controller
$R(k)$	Reference queue occupancy
$D(k)$	Independent frequency input to the network

peak power consumption constraints in interconnection networks by controlling the local power consumption of each router.

An on-line DVFS algorithm based on proportional-integral-derivative (PID) controller for multiple clock domain processors is presented in [16]. PID control requires *manual* tuning of the control gain which may become prohibitive as the number of voltage-clock domains increases. Moreover, PID control is useful for single-input-single output systems, so a coordination mechanism is needed in order to use it for controlling multiple queues across a network [5].

To address these shortcomings, we develop a state-space model for multiple clock domains and take the advantage of tools from modern control theory. More precisely, we propose:

- A state-space model for multiple voltage-frequency island NoC design, and analysis of the system controllability and stability
- Feedback control algorithms that control the speed of the voltage-frequency islands to cope with the PVT and workload variations.

## 3. NoC CONTROL METHODOLOGY

### 3.1. State-space Clock Domain Model and Notation

The utilization of any queue  $i$  at time  $t$  is denoted by  $q_i(t)$ , as shown in Table 1. Since the voltage and frequency values cannot be changed instantaneously, we define a control interval  $T$  during which the operating values are fixed, e.g., the  $k^{\text{th}}$  control interval is given as  $[kT, (k+1)T]$ .

We denote the utilization of the queue at the beginning of the  $k^{\text{th}}$  control interval by  $q_i(k)$ . The average arrival and service rates for the  $k^{\text{th}}$  interval associated with queue  $i$  are denoted by  $\lambda_i(k)$  and  $\mu_i(k)$ , respectively. Therefore, the dynamics of a single queue can be written as [16]:

$$q_i(k) = q_i(k-1) + T(\lambda_i(k-1) - \mu_i(k-1)) \quad (1)$$

We note that the occupancy of finite queues is bounded from below by zero, and from above by the queue capacity. So, Equation 1 represents the linearized dynamics around an operation point specified by the reference inputs.

We show next that the closed loop system can be stabilized around this operating value. We assume that the average arrival and service rates are proportional to the clock frequency<sup>1</sup>, so we have:

$$\lambda_i(k) = \bar{\lambda}_i \times f_1(k), \quad \mu_i(k) = \bar{\mu}_i \times f_2(k)$$

where  $f_1(k)$  and  $f_2(k)$  are the clock frequencies, while  $\bar{\lambda}_i$  and  $\bar{\mu}_i$  are the average *write* and *read* operations per cycle at the input and out-

1. If this relation does not hold, linearizing feedback as in [16] can be used.

put of the interface queue, respectively. Therefore, we can rewrite Equation 1 as follows:

$$q_i(k) = q_i(k-1) + T \begin{bmatrix} \bar{\lambda}_i & -\bar{\mu}_i \end{bmatrix} \begin{bmatrix} f_1(k-1) \\ f_2(k-1) \end{bmatrix} \quad (2)$$

Equation 2 describes the dynamics of a *single* queue that connects *two* different VFIs, as in Figure 2(a). Since, in general, in a system there exist *multiple* islands, more than one queue needs to be controlled. Hence, we define the state of the network with  $N$  interface queues, at time  $k$ , as:

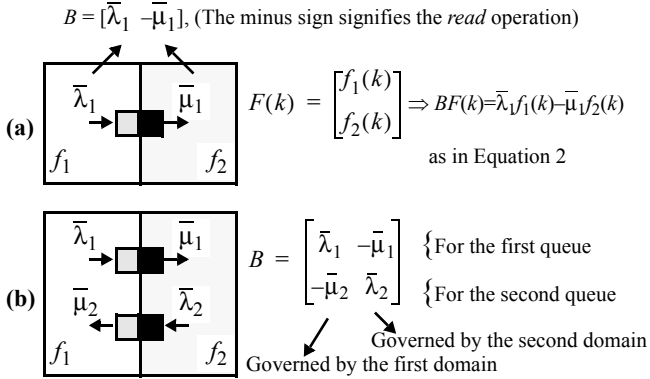
$$Q(k) = [q_1(k), q_2(k), \dots, q_N(k)]^T$$

For a system with  $M$  VFIs and  $N$  queues, the state-space of the *collective* queue dynamics is given by:

$$[Q(k)]_{N \times 1} = [Q(k-1)]_{N \times 1} + T [B]_{N \times M} [F(k-1)]_{M \times 1} \quad (3)$$

In this equation, the  $M \times 1$  vector  $F(k)$  is the control input where the  $i^{\text{th}}$  entry gives the frequency of the clock domain  $i$ . The  $N \times M$   $B$  matrix, on the other hand, describes the average *read* and *write* operations to the queue based on the system topology.

**Illustration of the modeling approach:** Matrix  $B$  in Equation 3 has one *row* for each state, and one *column* for each control input. The  $(i,j)^{\text{th}}$  entry of  $B$  is the rate of *write* (*read*) operations at the input (output) of the  $i^{\text{th}}$  queue due to the activity in the  $j^{\text{th}}$  voltage-frequency island. So, each row has at most two nonzero entries. For example, in Figure 2(a),  $B$  is a  $1 \times 2$  matrix:  $B(1,1)$  gives the rate at which island 1 writes to the queue, while  $B(1,2)$  is the rate at which island 2 reads from the queue. For this system,  $B = [\bar{\lambda}_i \quad -\bar{\mu}_i]$ , as illustrated in Figure 2(a). A larger example with two VFIs and two queues is illustrated in Figure 2(b).



**Figure 2 Illustration of the modeling framework with two VFIs when there are (a) one and (b) two interface queues.**

### 3.2. Controllability of Queue Dynamics

We first analyze the state-space model given in Equation 3 and derive the conditions under which the system is controllable. Referring to Figure 1(b), these conditions determine which queues can be controlled for a given VFI configuration. As explained in Section 3.1, the number of interface queues ( $N$ ) determines the dimensions of the state vector  $Q$ , while the number of VFIs ( $M$ ) gives the number of control inputs, assuming that the operating voltage and frequency of each island can be controlled independently. We note that in general, there may exist more interface queues than islands ( $N \geq M$ ), as shown in Figure 1(b). Our first original result determines the *maximum* number of queues that can be controlled in a network with  $M$  islands:

**Theorem:** In the multiple voltage-frequency island system with  $M$  islands (*i.e.*,  $M$  *independently* controllable clocks) described by Equation 3, utilization of at most  $M$  queues at the interface of VFIs can be controlled and the system is controllable *iff*  $\text{rank}(B) = N$ .

**Proof:** The proof follows from the analysis of the model in Equation 3; namely, the controllability matrix  $U$  for this system is:

$$U = [B | IB | I^2 B | \dots | I^{N-1} B]$$

where  $I$  is the  $N \times N$  identity matrix. The rank of  $U$  is obviously equal to the rank of  $B$  which is of size  $N \times M$ :

$$\text{rank}(U) = \text{rank}(B) \leq \min(M, N)$$

We know that the system is controllable *iff*  $\text{rank}(U) = N$  [11]. So, the system can be *state controllable* only if  $N \leq M$ , *i.e.*, the number of queues under control is less than or equal to the number of VFIs.

Furthermore, since  $\text{rank}(U) = \text{rank}(B)$ , the system is *state controllable iff*  $\text{rank}(B) = N$ .  $\square$

As a result, we can only control  $M$  out of  $N$  interface queues. In fact, the reduced order system with  $M$  queues represents a controllable subspace. Since  $M$ , *i.e.*, the number of VFIs is expected to be small [10,14], this also implies a low controller complexity and implementation overhead, as discussed in Section 4.3. In the remainder of this paper, we focus on this controllable subspace. Whenever there exists more than one interface queue between two VFIs, we choose to control the one with heavier traffic, since this has the highest impact on power consumption and performance. However, we note that the proposed controllers are independent of this choice provided that  $B$  is of full rank; so the technique is general.

### 3.3. The Regulator Synthesis Problem

In this section, we consider the regulation problem where the nominal operating voltage and frequency for each island,  $(V_i, f_i)$  for  $i=1 \dots M$ , are already determined (off-line or on-line) by another algorithm such as [10]. For instance, this could be done during partitioning of the network into distinct VFIs to optimize the energy consumption while balancing the workload. Since this optimization depends on nominal parameter values known at design time, this may, in fact, result in a performance mismatch between different islands; *e.g.*, one of the VFIs may end up working unnecessarily fast. Our goal here is to design a regulation mechanism such that the voltage and frequency values are *dynamically-controlled around these nominal values* as follows:

$$V_i(k) = V_i + \Delta V_i(k), \quad f_i(k) = f_i + \Delta f_i(k)$$

The block diagram of the control system used to achieve this goal is depicted in Figure 3(a). In this diagram,  $R(k)$  is an  $N \times 1$  vector which denotes the reference utilization of the controlled queues. Our goal is to find the  $M \times N$  gain ( $K_0$ ) and state feedback ( $K$ ) matrices that will place the eigenvalues of the closed loop system inside the unit circle and hence stabilize the system around the reference points.

#### 3.3.1. Controller Design

We can rewrite Equation 3 by setting  $F(k) = K_0 R(k) - K Q(k)$ <sup>1</sup>:

$$Q(k) = (I - TBK)Q(k-1) + TBK_0 R(k) \quad (4)$$

The system described by Equation 4 is asymptotically stable around the set point if and only if, the eigenvalues of  $(I - TBK)$  are within the unit circle [11]. Moreover, a steady-state analysis reveals that  $K_0$  should be set equal to  $K$  so that the steady-state queue utilizations are equal to the set point given by  $R(k)$ .

1. With some manipulations, average utilization can be also used as feedback.

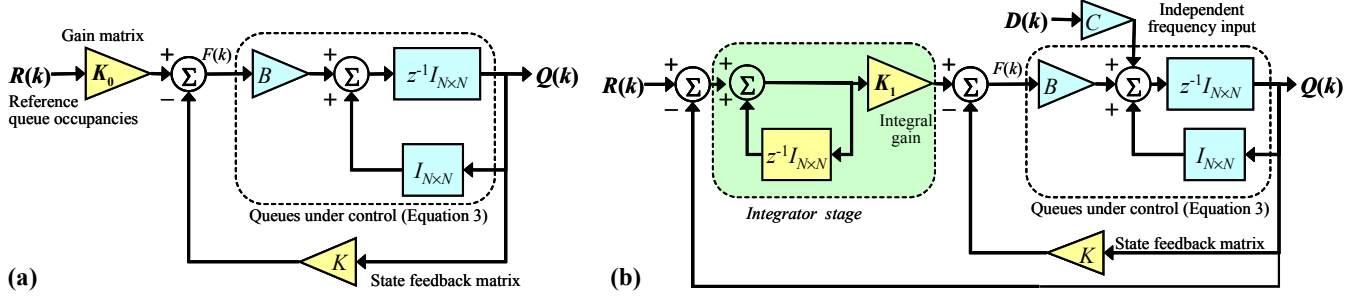


Figure 3 Block diagrams for the control systems for (a) regulation and (b) tracking problems are shown.

The state formalism brings the opportunity to use a rich set of control techniques to determine the state feedback matrix  $K$ . For instance, we can set  $K$  to place the eigenvalues of  $(I - TBK)$  to the desired locations inside the unit circle. We can also choose  $K$  using the so called *linear quadratic regulator (LQR)* to minimize a performance index ( $J$ ):

$$J = \sum_{k=0}^{\infty} [Q^T(k)GQ(k) + F^T(k)HF(k)] \quad (5)$$

where  $G$  and  $H$  are positive definite matrices that weight the relative importance of the state and control variables, respectively. For example, the  $H$  weights can be selected to prohibit large overshoots in frequency, hence voltage, and to minimize power consumption.

### 3.3.2. Further considerations

The techniques developed herein target on-chip implementations and so assume that the controller parameters are computed off-line to minimize controller overhead. Further robustness to parameter variations can be achieved by considering on-line gain scheduling techniques or off-line computation of feedback matrices for various operating conditions.

## 3.4. The Tracking Problem

The system studied in Section 3.3 is homogeneous, in the sense that all local frequencies are controlled by the proposed controller. In general, the frequency of one or more VFIs can be fixed to satisfy some given constraints. For example, input to an encoder system (or output of a decoder system) is set to satisfy an encoding (decoding) rate. Our goal for this scenario is to control the voltage and clock frequency of the remaining VFIs, so as to minimize the power consumption, while maintaining the reference queue utilizations.

The diagram of the control system is depicted in Figure 3(b). In contrast to the regulation case discussed above, in this case there exists an *external input* ( $D(k)$ ) which captures the frequency values that are set independently. Again, to provide some intuition, we start with a two-clock domain network with a single interface queue. The state-space model can be written as:

$$q(k) = q(k-1) - T\bar{\mu}_1 f_2 + T\bar{\lambda}_1 f_1 \quad (6)$$

Assume that the frequency of the first VFI (*i.e.*,  $f_1$ ) is fixed. We need to control  $f_2$  and  $V_2$  to minimize the power consumption, while maintaining the reference queue utilization. Intuitively,  $f_2$  should follow  $f_1$  such that the queue utilization is stabilized. In general, when there are  $M1$  VFIs under control and  $M2$  independent VFIs, the state-space model becomes:

$$Q(k) = Q(k-1) + TBF(k-1) + TCD(k-1) \quad (7)$$

where  $[F(k)]_{M1 \times 1}$  and  $[D(k)]_{M2 \times 1}$  represent the controlled and external frequencies, respectively. Likewise, the matrices  $B$  and  $C$

describe how the  $F(k)$  and  $D(k)$  affect the queue dynamics, as in the previous case. We also note that, for the simple system described by Equation 6,  $B = -\bar{\mu}_1$  and  $C = \bar{\lambda}_1$ .

We first show why the controller developed in the previous section cannot be directly used to regulate the queue utilizations in the presence of such external frequencies. For this purpose, we take the  $z$ -transform of Equation 7 to compute the transfer function from  $D(k)$  to  $Q(k)$  assuming that  $F(k) = -KQ(k)$

$$Q(z) = (zI - I + TBK)^{-1} TCD(z) \quad (8)$$

When the input frequencies are described by a step input,  $D(z) = z/(z-1) \times I_{M2 \times 1}$ , where  $I_{M2 \times 1} = [1, 1, \dots, 1]^T$  is a  $M2 \times 1$  column vector. By the final value theorem [11]:

$$\lim_{k \rightarrow \infty} Q(k) = \lim_{z \rightarrow 1} (1 - z^{-1})Q(z) = (BK)^{-1} CI_{M2 \times 1} \quad (9)$$

Thus, the external inputs *do* affect the steady-state queue utilizations. To make the queue utilization independent of the external frequency, we add an integrator stage in the front of the state feedback controller, as highlighted in Figure 3(b). Intuitively, the integrator adds a  $(z-1)$  term to  $Q(z)$ , so the right hand side of Equation 9 vanishes as  $z \rightarrow 1$ . The addition of the integrator doubles the number of states, since the error term for each state is integrated. Consequently, the original system needs to be augmented with new states. We note that the controllability of the original system is a *sufficient* condition for the controllability of the augmented system [11]. Then, the state feedback matrix  $K$  and gain matrix  $K_1$  can be computed using the techniques discussed in Section 3.3 to track the changes workload variations and input frequency, thereby achieving significant power savings.

### 3.4.1. Robustness to workload variations

To illustrate the impact of workload variation on the controller design, we consider a simple network with two VFIs. Suppose the first VFI operates at a fixed frequency,  $f_1$ , and determines the workload to the second VFI. We consider the following simple closed loop system:

$$q(k) = (1 - T\mu K)q(k-1) + T\lambda f_1 \quad (10)$$

The stability conditions is given as  $-1 < 1 - T\mu K < 1$ . However, if the nominal service rate  $\mu$  can vary by  $\Delta\mu$ , we should write:

$$\begin{aligned} -1 < 1 - T(\mu + \Delta\mu)K < 1 \\ 0 < K < 2/(T(\mu + \Delta\mu)) \end{aligned} \quad (11)$$

Consequently, selecting  $K$  based on the highest possible service rate guarantees stability in the presence of variations of the nominal values. We note that the PID based controllers [11] *do not* provide a systematic approach to deal with such variations and they would need manual tuning even for this simple system.

It is also possible to answer the converse question, *i.e.*, for a given  $K$ , how much variation can the system tolerate before losing stability? To answer this question for this example, we reorganize Equation 11 as follows:

$$-\mu < \Delta\mu < 2/(TK) - \mu \quad (12)$$

This equation implies that a negative variation whose amplitude is less than  $\mu$ , and a positive variation with amplitude less than  $(2/TK - \mu)$ , are acceptable. In general, for systems of larger dimensions, the feedback matrix  $K$  should be selected such that the stability condition is satisfied for a *range* of parameter variations.

## 4. EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of the proposed control techniques on several hardware designs. The interface queues we control are block RAM-based mixed-clock FIFOs from the Xilinx library [18]. Voltage conversion is also performed at the VFI interfaces; we assume 0.9 efficiency and  $10\mu\text{F}$  load capacitance for voltage conversion to compute the switching overhead as in [16]. Whenever slowing down a clock domain, first the frequency is set to the desired value and then the voltage is scaled down. Similarly, the voltage is ramped up first when speeding a domain up. Since, the voltage transitions may take more than  $10\ \mu\text{sec}$  [4,17], we conservatively set the control interval,  $T \approx 100\ \mu\text{sec}$ ; for implementation purposes, the exact value is selected such that  $T$  is a power of two times the slowest clock in the system.

### 4.1. Robustness to Parameter Variations

In the first set of experiments, we evaluate the performance of the proposed controllers in the presence of parameter variations and bursty *write* and *read* operations. When there is no closed loop controller, even a small deviation from the nominal frequency values results in fully utilized or empty interface queues, *i.e.*, a performance mismatch. On the other hand, the proposed regulation scheme successfully stabilizes the queue utilization under different application scenarios, as described below.

**Comparison with PID controller:** We first design a state-space based controller for the system in Section 3.4.1 (described by Equation 10). We verified both theoretically (using Equation 12) and experimentally that the closed loop system is stable in presence of 80% variation in the nominal value of the service rate  $\mu$ . On the other hand, a manually tuned PID controller for the same system becomes unstable for as low as 10% variation in service rate.

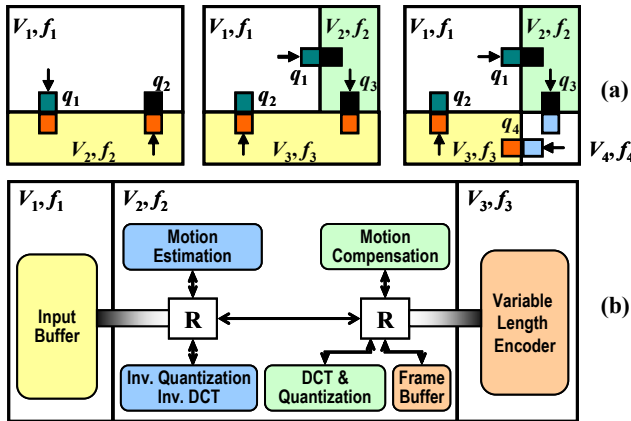


Figure 4 NoCs studied in (a) Section 4.1 and (b) Section 4.2.

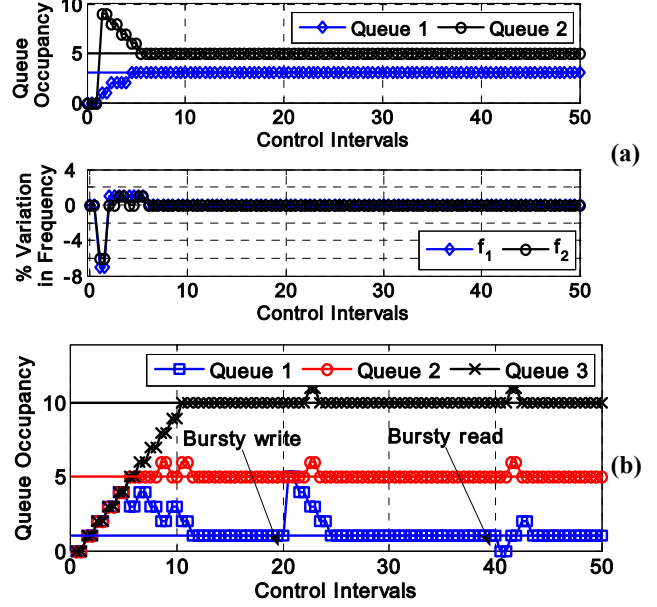


Figure 5 (a) Queue occupancy and variation in the input frequency for the NoCs with (a) two and (b) three VFIs.

**Regulator with a larger number of islands:** We also consider networks with two, three, and four islands shown in Figure 4 and design regulators as explained in Section 3.3. We design the regulator system for the two-clock domain network to obtain a fast response with less than 10% overshoot around the nominal frequency. For the resulting regulator system, we identify the maximum permissible variation in the arrival and service rates for the closed loop system as being about 20% of the nominal values. Then, the simulation is repeated by varying the arrival and service rates of both domains by 20% around the nominal value. As shown in Figure 5(a), we observe that the controller indeed stabilizes the network successfully within six control intervals.

**Bursty read/write operations:** Similarly, simulations are performed for the three-clock domain network shown in the center of Figure 4(a). In this case, we also inject (eject) bursty data to (from) queue  $q_1$  in the 20<sup>th</sup> (40<sup>th</sup>) control interval, as shown in Figure 5(b). We observe that this sudden change in the utilization of  $q_1$  is rapidly pushed back to the reference point. It is also interesting to note that in order to reduce the utilization of  $q_1$ , the second VFI in Figure 4(a) has to operate faster. This results in a smaller spike in the utilization of  $q_2$ , and eventually  $q_3$ , during the 22<sup>nd</sup> control interval. Finally, we obtain similar results for the 4-domain network in Figure 4(a); they are not shown here due to space limitations.

### 4.2. Experiments with MPEG-2 Encoder

The second set of experiments demonstrate the application of the tracking controller to an NoC-based hardware implementation of an MPEG-2 encoder presented in [6]. We divided this design into three domains (Figure 4(b)). The first domain contains the input buffer and related control logic which prepares the macroblocks to be read by the discrete-cosine transform (DCT) and motion estimation (ME) modules. The third domain contains the run-length encoder, Huffman encoder and zig-zag modules, while the second domain contains the remaining cores. We compare the power consumption of this design against the power consumption of a baseline design which has only a single clock domain.

The encoder system needs to meet a target frame rate of 50 Frames/sec for 352×288 CIF frames. The baseline design has to operate at 100MHz to achieve this frame rate. Direct measurements on the hardware reveal that this system has an average power consumption of 1.90W. The minimum frequency for the single clock domain system is dictated by the ME and DCT modules. In order to achieve the target throughput with the design with three VFIs, we set the frequency of the first island ( $f_1$ ) such that it will accept the raw frames and prepare data for the ME, MC and DCT modules at the target rate of 50 Frames/sec; that is, in terms of the block diagram shown in Figure 3(b), the independent frequency is  $f_1$ , *i.e.*,  $D(k)=f_1$  in Equation 7. Once  $f_1$  and the reference utilizations for the queues are set, the frequency of the second domain ( $f_2$ ) is set *automatically* by the controller to match to the speed of the first domain. Likewise, the controller adjusts the frequency of the third domain ( $f_3$ ) accordingly. We note that when the domains operate at their nominal workloads,  $f_2$  is indeed set to 100MHz.

Obviously, we can achieve about 10% power savings by operating the first and third domains at lower frequencies without affecting the overall performance. However, the real savings are obtained due to the variation in the *workload*. It has been observed that there is significant variation in the contents (hence size and processing time) of different macroblocks [15]. So the workload of the second domain varies significantly for each macroblock. As a result, the controller can dynamically lower the operating frequency whenever needed. Indeed, the frequencies of the second and third VFIs adjust to the workload changes, as we show in Figure 6. As a result of adjusting the frequency, about 20% power savings can be achieved. When we also scale voltage accordingly, power savings jump to about 46% *after* accounting for the DVS overhead.

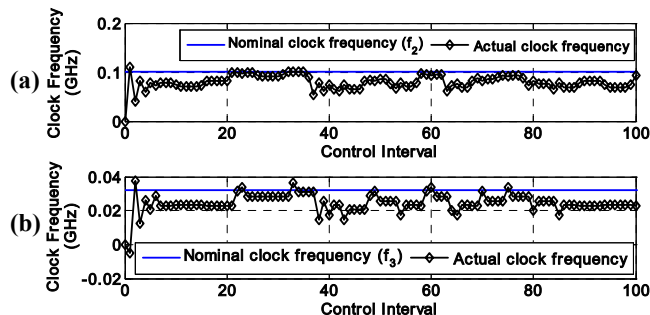


Figure 6 The frequency adopted by the second (a) and third (b) clock domain in response to workload variations, thereby saving energy.

### 4.3. Implementation of the Controller

Due to possible power distribution, clock generation and inter-domain communication issues, the number of islands is expected to be small [10,14] and so, the controller is expected to have low complexity. In term of the performance, the control interval is large enough (>10,000 cycles) for the feedback and control signals to reach their destination domains. These signals may also have a high priority or use a network protocol that guarantees on-time delivery. So, the time needed for computation and communication of control signals are not critical.

To evaluate the area overhead, we implement an FPGA prototype using Xilinx Virtex-II Pro<sup>®</sup> XC2VP30. Four basic clocks are generated using delay locked loops available in the FPGA device. The clock control module which implements the proposed control algo-

rithm is able to drive 22 different clock signals using this basic clock signals. Our current implementation uses PicoBlaze micro-processor for flexibility reasons [18]. Voltage level conversion is not supported by the current prototype, since it is not available for the Xilinx platforms. In general, we assume that each island will generate its frequency from a common PLL and have its own voltage regulator as described in previous studies [8]. Finally, the area of the controller current FPGA implementation corresponds to less than 3% of the MPEG-2 encoder discussed in Section 4.2.

## 5. CONCLUSION

In this paper, we presented a variation-adaptive feedback control methodology for NoC architectures with multiple voltage-clock domains. More precisely, we developed techniques to dynamically control the speed of each VFI and provide robustness against the uncertainties and variations in the design parameters. At the same time, our techniques adapt the operating voltage and frequency to the variations in the workload to save power. Experiments demonstrate robustness to parameter variations and more than 40% energy savings for a hardware MPEG-2 encoder design.

**Acknowledgements** This work is supported in part by GRC grant 2008-HJ-1800, NSF Awards CCF-0702420 and NS-0720529. The authors also acknowledge the GSRC support for related work that lead to the ideas presented in this paper.

## 6. REFERENCES

- [1] S. Borkar, *et al.* "Parameter variations and impact on circuits and microarchitecture," In Proc. DAC, June 2003.
- [2] T. Chelcea, S.M. Nowick, "A low latency fifo for mixed-clock systems," In Proc. IEEE Computer Society Workshop on VLSI, April 2000.
- [3] W. Dally, B. Towles, "Route packets, not wires: On-chip interconnection networks," In Proc. DAC, June 2001.
- [4] Intel Corp. Enhanced Intel<sup>®</sup> SpeedStep<sup>®</sup> Technology for the Intel<sup>®</sup> Pentium<sup>®</sup> M Processor. March 2004.
- [5] P. Juang, *et al.*, "Coordinated, distributed, formal energy management of chip multiprocessors," In Proc. of the ISLPED Aug. 2005.
- [6] H. G. Lee, *et al.*, "On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches. ACM TODAES, vol.12, no.3, Aug. 2007.
- [7] M. Liu, *et al.*, "Leakage power reduction by dual-vth designs under probabilistic analysis of Vth variation," In Proc. ISLPED, Aug. 2004.
- [8] G. Magklis, *et al.*, "Independent front-end and back-end dynamic voltage scaling for a gals microarchitecture, In Proc. ISLPED, Oct. 2006.
- [9] S. Martin, *et al.*, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," In Proc. ICCAD, Nov. 2002.
- [10] U. Y. Ogras, *et al.*, "Voltage-frequency island partitioning for GALS-based networks-on-chip, In Proc. DAC, June 2007.
- [11] K. Ogata. Discrete-Time Control Systems. Prentice-Hall, Upper Saddle River, NJ, 1995.
- [12] L. Shang, L. Peh, N. K. Jha, "PowerHerd: Dynamically satisfying peak power constraints in interconnection networks," In Proc. of the Intl. Symp. on Supercomputing, June 2003.
- [13] T. Simunic, S. P. Boyd, P. Glynn, "Managing power consumption in networks on chips," In Trans. on VLSI, vol.12, no.1, Jan. 2004.
- [14] The international technology roadmap for semiconductors, 2006.
- [15] G. Varatkar, R. Marculescu, "On-chip traffic modeling and synthesis for MPEG-2 video applications," In IEEE Trans. on VLSI, vol.12, no.1, Jan. 2004.
- [16] Q. Wu, *et al.*, "Formal online methods for voltage/frequency control in multiple clock domain microprocessors," In Proc. ASPLOS, Oct. 2004.
- [17] F. Xie, M. Martonosi, S. Malik, "Intraprogram dynamic voltage scaling: Bounding opportunities with analytic modeling," ACM Transactions on Architecture and Code Optimization, vol.1, Issue 3, Sept. 2004.
- [18] Xilinx IP Library. [Online] <http://www.xilinx.com/ipcenter/index.htm>