Design Space Exploration of DVFS on On-Chip Hybrid Communication Networks

Carlos Gabriel de Araujo Gewehr$^1$ and Mateus Beck Rutzig$^1$

$^1$ Electronics and Computing Department, Federal University of Santa Maria, Santa Maria, Brazil
e-mail: mateus@inf.ufsm.br

Abstract— Multi-processor systems-on-chip (MPSoCs) have been established as the standard platform for high-performance applications in the semiconductor industry. With an increasing number of Processing Elements (PEs) in a single die, communication scalability is one of the foremost concerns. Networks-on-chip (NoCs) have already been shown to be an alternative to provide scalability, but they rely on a homogeneous communication fabric, which is not ideal for the heterogeneous communication demand of current embedded devices. Hybrid communication structures can mitigate such issues by merging different communication structures that can better accommodate heterogeneous communication profiles. In this work, a Hybrid Communication Infrastructure is proposed by merging fully-, partially- and peer-to-peer connected fabrics providing a wide range of communication bandwidth. In addition, we couple to the hybrid fabric a multi-grained DVFS approach that dynamically adapts voltage and frequency of the structures considering the application’s performance needs. Results show that the Hybrid structure achieves up to 22% power savings and occupies 42% less chip area than a NoC with the same number of PEs. Experiments with relevant video encoding applications show power savings of up to 70% over a homogeneous NoC, with no significant throughput losses, both coupled to the proposed DVFS approach.

Index Terms— Network-on-Chip (NoC); Dynamic Voltage and Frequency Scaling (DVFS).

I. INTRODUCTION

The integration of multiple IPs like processors, memories, and I/O interfaces in Systems-on-Chip (SoCs) poses challenges in terms of scalability and power and area consumption due to communication fabric design. The conventional interconnect designs are not suitable to address these concerns on a larger scale, since buses have significant power consumption and crossbars have significant area costs.

The scalable alternative to buses and crossbars is Networks-on-Chip (NoCs), but they still come with significant power and area costs [1, 2]. However, NoCs and other homogeneous communication structures are not flexible in terms of bandwidth range. Hybrid network topologies offer an opportunity for power and area savings as modern SoCs have a wide range of throughput demands.

Dynamic Voltage and Frequency Scaling (DVFS) is a widely-used technique for reducing power consumption in NoCs. It is an attractive option because it allows establishing voltage and frequency domains from router to global granularity. This way, specific routers can have their performance fine-tuned, enabling near-optimal power consumption with such precise adjustments. However, decentralized control is required for this fine-grained adjustment, which can be a drawback. Alternatively, global tuning simplifies the control logic required, but results in smaller power savings.

Despite using DVFS, power savings can still be limited when implementing a Network-on-Chip (NoC) due to area constraints. Hybrid communication infrastructures have been developed to address both power and area issues while maintaining scalability. By integrating different communication techniques in a hybrid architecture, a wide range of communication bandwidth can be achieved at varying power and area costs. Additionally, with DVFS, further power savings can be obtained by adjusting the voltage and frequency settings of network components to match an application’s expected throughput requirements, at the cost of increased die area usage.

The contributions of this work come in two directions:

• First, a hybrid, transparent packet-based on-chip communication infrastructure is presented. Due to its heterogeneous topology, the proposed infrastructure provides more flexibility to achieve a better matching between the demanded application throughputs and network-provided bandwidth. For instance, threads of an application that have a low communication demand may be clustered together into peer-to-peer structures, allowing for sufficient performance at low power and area costs. On the other hand, threads with high communication demands may be placed within the fully connected network, allowing for performance to be adequately provided, albeit at higher power and area costs.

• Second, we propose a Dynamic Voltage and Frequency Scaling (DVFS) design space exploration for the hybrid structure that considers three grains of voltage and frequency domains: router, structure, and global. The proposed DVFS approaches further improve the matching between an application’s threads demanded throughputs and their allocated structures provided bandwidth, since available bandwidths can be set by adapting, at run-time, the operating frequency of elements of the structure. For instance, if there is a thread that does not fit adequately in a Bus (peer-to-peer) or a Crossbar (fully connected), fine-grained DVFS may be leveraged to place this thread in a Router and set its frequency to provide exactly the throughput it needs, without compromising the performance of other threads in a Bus/Crossbar, and the power and area savings obtained from placing them at a Bus/Crossbar.
To investigate the Hybrid Structure and the suggested DVFS approach, we conducted RTL simulations of various Hybrid configurations using real-world application workloads from video encoding and video processing applications. We compared the Hybrid scenarios to a baseline NoC, with and without the proposed DVFS approach, in terms of power consumption, throughput, and latency to determine their effectiveness.

The document is organized as follows: Section 2 provides an overview of the core concepts related to on-chip interconnection networks. The section also discusses related works on Hybrid Structures and DVFS on NoCs. In Section 3, the Hybrid Structure is presented, and in Section 4, the DVFS implementation is discussed. Section 5 provides a detailed experimental evaluation of the Hybrid Structure and the DVFS implementation. Finally, Section 6 presents final remarks and suggestions for future works.

II. BACKGROUND & RELATED WORKS

Many works in the recent past have explored hybrid network topologies and DVFS in the context of NoCs to reduce power consumption. In this chapter, these works and their proposed policies and strategies for determining Voltage and Frequency (VF) pairs are summarized and classified by their network/system-level characteristics.

A. Hybrid topologies in interconnection networks

The simplest hybrid topology interconnection optimization is shown in [3]. The authors present a hierarchical approach: PEs are first clustered in Buses, based on the communication affinity between them. Clusters are then interconnected through a 2D mesh NoC. By keeping Bus clusters small, the Bus scalability tipping point is never reached. Required Throughputs can be maintained, while offering attractive performance-per-watt. If clusters are large enough, the NoC’s severe initial power and area costs also can be reduced, seeing as fewer Routers in the 2D mesh NoC will be needed. Experiments with video encoding applications demonstrate reduced latency and 44% less area when compared to a homogenous 2D mesh NoC. Unfortunately, no power consumption analysis was provided.

Similar proposals are presented in [4] and [5]. [4] presents a concrete clustering algorithm, left abstract in [3], which allows for the optimal grouping of application threads for a given Bus-NoC hybrid topology. Protocol heterogeneity is also explored, using AHB-Lite for the Bus and a Credit-based protocol for the NoC, similar to the one in Hermes (used in this work). A reduction of up to 24% in packet latency was observed, but again, no power consumption analysis was provided.

[5] also presents a mapping algorithm for allocating PEs in a regular hybrid network topology. The topology in question is a 2D mesh NoC, with a Distributed Time Division Multiple Access (dTDMA) Bus of varying size at each Router local port. Applications are mapped to PEs to minimize inter-cluster communication. In a comparison with a homogeneous 2D mesh NoC, it shows a power consumption decrease of up to 69% and up to 31% decrease in average packet latency.

In [6], the authors argue that NoCs have a clear benefit over the traditional Bus in scalability and parallelism, while a Bus is superior in terms of latency and multicast communication. Thus, a hybrid interconnect, containing both a mesh 2D NoC and a global Bus makes itself attractive. By sending high-throughput unicast communication through the NoC and low-throughput multicast communication through the Bus, both power and performance can be optimized.

Power consumption is reduced due to the lesser amount of replicated unicast messages to be sent through the NoC, emulating multicast functionality. This increases NoC performance by reducing Link congestion, again, due to the reduction of the total amount of packets traveling in the NoC. The proposal is evaluated in a Dynamic Non-Uniform Cache Access (DNUCA) multiprocessor system, where the desired communication behavior can be observed. Results show an average of 32% application execution time decrease while showing an average of 18% energy savings.

Another relevant proposal for using Buses in a NoC context is presented in [7]. Instead of linking PEs to either a Bus cluster or directly to the NoC, as done in the works above, PEs are connected to both a Router and a Bus cluster. PEs in the same Bus cluster communicate through it, while PEs in different clusters communicate through the NoC. This results in the reduction of network congestion in the NoC, again reducing packet latency, on average, by 40%, when compared to a simple 2D mesh NoC.

B. Dynamic Voltage and Frequency Scaling in NoCs

B.1 Bandwidth and Throughput Definitions - A link’s Bandwidth $B$ is defined by the maximum amount of bits that can be transmitted per second. In the present context, it can be quantified through Equation 1, where $DW$ is the data width of the link, in bits, and $f$, the clock frequency.

$$B = DW \ast f$$  \hspace{1cm} (1)

Some issues, such as network congestion, can avoid reaching the Bandwidth, which is the ideal transmission value of a specific structure. Also, there are situations where the entire Bandwidth of the link is not used by a certain application. So, the effective value for information traveling through a link is called Throughput. Since it is a fraction of a link’s Bandwidth, the Throughput $T$ is also measured in bits per second. Equation 2 quantifies the Throughput of a link, where $R$ is a ratio (between 0 and 1) used by the Bandwidth provided by the link in question.

$$T = R \ast B$$  \hspace{1cm} (2)

B.2 Network-level DVFS policies - The authors of [8] present the fundamental network-level policy for DVFS in NoCs. By the use of Throughput evaluating modules at each router’s ports, these modules set their associated router’s clock frequency to the lowest possible frequency that can provide the required throughput (as evaluated by the aforementioned modules). Supply voltage is set to the lowest possible voltage (within a range of discrete voltage values, in the experiment performed by the authors, 2) that can sustain the router’s operation at the determined clock frequency.
By matching the network-provided throughput to the application’s required throughput, significant power savings are obtained, at the cost of average message latency. When compared to a theoretical global DVFS approach, where the whole NoC is subject to a single VF (Voltage-Frequency) pair; and a theoretical ideal local DVFS, where supply voltage levels are not discretized, the author’s proposal consumes, on average, 33% less than the global approach.

In [9], the authors formalize the previously described throughput matching policy as Rate-based Max Slow Down (RMSD), and introduce a novel metric, Delay-based Max Slow Down (DMSD), where, instead of establishing a minimum throughput, as in the RMSD metric, a maximum message latency value is established, and VF pairs are set so messages do not have a higher latency than a target maximum. These two policies, as well as a No-DVFS scenario, are then compared in terms of power consumption and message latency, while running benchmark applications. In those, compared to the RMSD metric, the DMSD policy presents, on average, 2 times less latency and 1.4 times less power consumption.

In [10], the authors extend their previous work, mentioned above (RMSD vs DMSD), while formalizing a third metric, Queue-based Max Slow Down (QMSD), where VF pairs are set based on First-In-First-Out (FIFO) queue occupancy, increasing the clock frequency (and by extension, Throughput) of a router as its buffers get increasingly filled. The three policies (RMSD, DMSD and QMSD), as well as a No-DVFS scenario, are compared in a broader set of benchmarks, in terms of power consumption, message delay and power-delay product. The RMSD and QMSD policies have very similar observed behavior. Between the three policies, the results obtained show better power figures for the RMSD/QMSD policies, but better delay figures for the DMSD policy. As for the power-delay product, the RMSD/QMSD policies present slightly lower values, signaling a better power-performance trade-off when compared to the DMSD policy.

The authors of [11] implement a prediction-based DVFS scheme, where Buffer Utilization (BU) and Link Utilization (LU) histories are used in predicting the future usage state of the router, taking a proactive stance in determining VF pairs, as opposed to the works cited so far, which react to the perceived changes in network/router state, and only then make a DVFS decision. For each input buffer and associated link in a router, at the end of a set time window, its own BU and LU are evaluated for the current time window, and used in estimating the BU and LU of its associated downstream buffer/link for the next time window. Averaging out the estimated downstream BU/LUs, a router’s VF pair is set accordingly (For the sake of simplicity, decreased if high usage is to be expected, or increased if low usage is to be expected). Power and delay results are presented for the proposed technique with increasing time window sizes and compared to a nominal No-DVFS scenario, showing significantly less power consumption in the proposed technique, increasing with time window size. Delay figures show a small decrease in cases where the time window is a close match to packet sizes, and a small increase otherwise.

A more solid evaluation of the aforementioned proposal is presented in [12], where instead of synthetic traffic, the NoC and its DVFS controllers are subject to a real application, an H.264 encoder. In this case, the authors report, on average, around 30% power savings as well as around a 4% latency decrease when compared to a baseline No-DVFS scenario. [13] uses the same MPSoC of the presented work proposing a DFS approach both in the PEs and NoC structure. The DFS controller uses the packet information to decide the operating frequency of individual routers. For the PEs, the operating frequency is set considering the computation and communication load. The proposal achieves power savings of 26% in the PEs side and 76% in the NoC side.

### B.3 System-level DVFS policies

In [14], the authors propose a distributed approach to determining VF pairs, where each thread determines at run-time an ideal VF pair for the router cluster it is allocated to. The VF pairs determined by each thread are counted as votes in a pool of possible VF pairs, and the final VF pair for each cluster is then chosen as the one that received the most votes out of the possible VF pairs in the voting pool. The proposal is evaluated in system-level simulations against two other scenarios: where no DVFS is performed, and with a policy very similar to [10]’s QMSD. In terms of Million Packets per Joule (MPpJ), the authors report savings of up to 17.9% in network energy consumption, and, in terms of Million Instructions per Joule (MiJp), up to 26.3% in system energy consumption.

The authors of [15] observe and discuss a non-linear relationship between power and performance in the context of NoC DVFS. In a normalized Power by Performance Characteristic Curve (PPCC), the authors identify three distinct regions: an Inertial region, where providing more power to the NoC doesn’t provide significant performance benefits, due to severe bottle-necking/congestion; a Linear region where application performance is linear to NoC power, due to the NoC being the main bottleneck on overall performance; and a Saturation region, where an increase in NoC performance doesn’t translate to an increase in application performance, due to the NoC not being the overall performance bottleneck.

Considering that, if the NoC is operating in either the Inertial or Saturation region, power is being wasted (which follows from the fact that power expenditure while in these regions doesn’t lead to significant application performance gains), a new metric for evaluating the power-performance trade-off is introduced. Marginal Performance (MP), accounting for potentially mislead DVFS decisions, if a linear power-to-performance relationship is to be assumed. With the MP metric, the NoC operates only in the Linear region of the PPCC, which leads to only meaningful (as in, leads to overall performance benefits) power expenditure. Through extensive system-level simulations, the authors show power under- and over-provisioning in two relevant NoC DVFS implementations (their previous thread voting work in [14], and one very similar to the QMSD policy in [10]) through the proposed PPCC-MP method, but no method for the real-time determination of VF pairs is formalized.

[16], a follow-up of the author’s previous work in [15] (mentioned above), demonstrates an implementation of NoC DVFS (Δ-DVFS) guided by their previously presented met-
ric of PPCC. In \(\Delta\text{-DVFS}\), a PE’s thread workload is continuously profiled by a monitor, and based on this profile, a target value in the PPCC is determined, which follows the picking of a VF pair that more closely matches the target value. The authors report averages of 38.9% power consumption reduction and 2.3% application execution time increase over the previous proposal using the same scenarios.

[17] presents a proactive DVFS approach in the context of shared memory Chip Multi-Processors (CMPs). This is accomplished through the predictable nature of cache coherence to set relevant VF pairs proactively, rather than wait for the network to observe a state change and, only then, react accordingly, as done in previously discussed works. The case for proactive DVFS is made through the observation that, from the point of view of a router or link, the network state can drastically change without warning. From this, it follows that a factual change in network state, as it cannot be reliably predicted, can only be perceived after a certain amount of time, in which VF pairs are set accordingly to a previous network state, leading to an obvious inefficiency. By removing this inefficiency associated with the time taken to observe a change in network state, proactive DVFS can lead to further power savings than reactive DVFS. The authors’ proposal shows a 41% power consumption decrease when compared to a No-DVFS scenario and, on average, a 21% decrease when compared to reactive DVFS.

C. Our contribution

In this work, we propose an exploitation of DVFS in a Hybrid Communication Structure that unlike:

- [5, 4, 3, 6, 7], it aggregates higher heterogeneity in the network by coupling, besides Buses, Crossbars to increase the range of available bandwidth, allowing for better a match from interconnect to application throughput requirements.

- [11, 10, 17, 15, 12, 8, 14, 16, 13], it goes a step further by providing a dynamic and parameterizable DVFS over different structures in a Hybrid topology, which besides matching the application throughput requirements to the ideal structure, tunes the operating frequency and voltage supply to boost power savings.

III. THE HYBRID INTERCONNECTION NETWORK

In this section, the design of the interconnection infrastructures (Bus, Crossbar and NoC) and further details on integrating them in a hybrid network topology are discussed.

A. Overview

A Hybrid interconnection can be achieved by combining Buses and Crossbars into a base NoC, as shown in Figure 1. The NoC used in this work is based on Hermes NoC [18], which is employed alongside Packet Switching Shared Buses and Packet Switching Crossbars. To transparently provide communication in the Hybrid structure, an entity, named Bridge was developed. The Bridge is essential for maintaining the wormhole and packet switching characteristics of the Hermes NoC, providing a transparent interface to PEs.

A packet switching interconnection means that the data is transmitted in the form of packets. Packets are divided into flits which are the smallest amount of data that can be transmitted in one link on each cycle. As shown in Figure 2, each packet has a header and a payload. The header comprises the first two flits of a packet, which contains the target address and the payload size, respectively. The remaining flits are the payload, the actual data sent to the receiving entity.

B. Network-on-Chip (NoC)

The NoC implementation used is a 2D packet-switched wormhole NoC with an XY routing algorithm [18]. In a wormhole-switched NoC, each flit is transmitted right after the previous, not waiting for any subsequent flits to arrive in the current router. The XY routing algorithm, first, routes a packet along the X dimension, moving from the sending PE’s X coordinate to the receiving PE’s X coordinate, but conserving its position in the Y dimension. After, the packet travels through the Y dimension, towards the receiving PE’s Y coordinate. The use of XY routing leads to low hardware implementation costs (due to low complexity, implying low power and area costs) and a deadlock-free guarantee (no recursive dependencies).

Router-to-router communication is illustrated in Figure 3. The inter-router protocol is a simple ready-valid handshake. The success of a single flit transfer is determined by two control signals: Tx (transmitter side) and Credit (receiver side). The Tx signal informs the receiver that there is valid data on the Data bus. The Credit signal informs the transmitter that the receiver can write a flit in its input First-In-First-Out (FIFO) buffer. If both signals are simultaneously asserted, the flit in the Data bus has been successfully written into the FIFO, and another flit can be placed in the Data bus.
C. Bus

Interaction with the Arbiter is done exclusively through an intermediary entity named Bridge, which is depicted in Figure 4. The Bridge is responsible for assuring transparent communication between PEs of different communication structures. As shown in Figure 4, the Bus bridge interfaces with the NoC Router (top and bottom interface) and the Arbiter (right interface).

We have implemented a Bus to follow the Hermes NoC’s packet-switched communication protocol as shown in Figure 5. The bus Arbiter follows the Round Robin algorithm, which aims for a fair distribution of Grants among PEs. The algorithm works by selecting a PE in a pool of initially equal-priority requesting PEs. After the selected PE has given up its Grant, by asserting the Bus’ ACK signal, its priority is set to the lowest possible. This process is performed at every ACK event, setting the corresponding PE’s priority to the lowest possible, and incrementing by one all other PE’s priorities. In this arbitration scheme, the lowest priority PE will always be one that most recently had access to the Bus, and the highest priority PE, is the one that least recently had access to the Bus. Once a Grant signal is asserted by the Arbiter, the source PE is defined, allowing that PE to write into the Data and Tx lines of the Bus.

D. Crossbar

The Crossbar Bridge also has the same structure as the Bus Bridge, but was modified to interact with \( N \) Arbiters instead of only 1, as shown in Figure 6. \( N \) comparators are used to compare the \( ADDR \) flit to known addresses of PEs in the Crossbar. The Crossbar follows the same structure and arbitration as the Bus, as shown in Figure 7.

E. Integration between Communication Structures

Aside from implementing the Bus and Crossbar following the packet-switching wormhole logic previously described, a modification of the PE addressing scheme must be performed. A Global Address is necessary to forward packets outside the NoC. So that after packets travel inside the NoC (using the Base NoC Address), they should be forwarded to the specific PE in a Bus/Crossbar (using the Global Address). For that, we have modified the Hermes addressing methodology. Figure 8 (a) shows an original Hermes address header flit and (b) the modified header flit. The Global Address is stored in the 16 higher order bits, since they are not used by the Hermes addressing protocol. In the 16 lower-order bits, the address of the PE inside the NoC is stored.

The Global Addressing behaves as follows. Suppose the hypothetical hybrid topology shown at the top of Figure 9. For PEs in the NoC, Global Addresses are taken as the position in such structure. For the remaining PEs (not in the base NoC) Global Addresses are assigned in the following manner: Sorted by their position in the base NoC, first Buses, then Crossbars, are superposed in the NoC in perimeters along the base NoC’s projection. As shown in the middle of Figure 9, each of these perimeters follows a clockwise rotation around the base NoC’s projection, starting from the left-most position and finishing when an edge of the NoC is reached. The final PE addressing is shown at the bottom of Figure 9.
IV. DESIGN SPACE EXPLORATION OF DVFS IN THE HYBRID TOPOLOGY

A. Overview

Previous works concerning the use of DVFS in NoCs present several possibilities in DVFS decision-making and granularities. In this work, we implemented:

- the DVFS decision-making entirely by software. Such an algorithm runs in the master PE, which performs task allocation and voltage and frequency decisions. When the decision is taken, the master PE sends a specific packet to the DVFS controller which sets the voltage and frequency.

- three DVFS granularities: Router-grained, where each Router, Bus and Crossbar have individual VF-pairs; Structure-grained, where each Bus, NoC and Crossbar have an individual VF-pair; and Global-grained, where a single global VF-pair is applied.

B. DVFS decision-making software

The proposed software-based DVFS runs in the master PE that handles the allocation of tasks in the system and the VF pair of every communication module. To find the VF pair of the communication modules the algorithm receives a task graph of the application that contains the Throughput of the communicating threads, as shown in Figure 10. Thus, for determining each Bus/Crossbar/Router’s frequency, the algorithm gathers from the Throughput of the network’s links. The minimum frequency required for a link to maintain a given Throughput can be obtained from Equations 1 and 2, setting $R$ as 1 (Bandwidth = Throughput):

$$f(T) = \frac{T}{DW}$$

(3)

The Bus’s Throughput $T_B$ is given by the sum of its input, output and local Throughputs ($T_{B_i}, T_{B_o}, T_{B_l}$, respectively), as depicted in Equation below.

$$T_B = T_{B_i} + T_{B_o} + T_{B_l}$$

(4)

The minimum Bus frequency can be determined from Equation 3:

$$f_B = \frac{T_B}{DW}$$

(5)

Considering a Crossbar with $n$ PEs, there is a set $T_{C_o}$ containing the output Throughputs at each PE $T_{C_{o_i}}$, and a set $T_{C_i}$ containing the input Throughputs of each PE $T_{C_{i_n}}$.

$$T_{C_i} = \{T_{C_{i1}} + T_{C_{i2}} + \ldots + T_{C_{in}}\}$$

(6)

$$T_{C_o} = \{T_{C_{o1}} + T_{C_{o2}} + \ldots + T_{C_{on}}\}$$

(7)

The Crossbar’s minimum frequency is obtained from the maximum Throughput in both input and output port Throughput sets, as defined in Equation 8:

$$f_C = \frac{\text{max}(\text{max}(T_{C_i}), \text{max}(T_{C_o}))}{DW}$$

(8)

The same applies to NoC Routers. Considering a Router with $n$ ports, there is a set $T_{R_o}$ containing the output Throughputs at each port $T_{R_{o_i}}$, and a set $T_{R_i}$ containing the input Throughputs at each port $T_{R_{i}}$.

$$T_{R_i} = \{T_{R_{i1}} + T_{R_{i2}} + \ldots + T_{R_{in}}\}$$

(9)

$$T_{R_o} = \{T_{R_{o1}} + T_{R_{o2}} + \ldots + T_{R_{on}}\}$$

(10)

As with a Crossbar’s, a Router’s frequency is also obtained from the maximum Throughput in both input and output port Throughput sets, as defined in Equation 11:

$$f_R = \frac{\text{max}(\text{max}(T_{R_i}), \text{max}(T_{R_o}))}{DW}$$

(11)
C. DVFS controller implementation

Voltages and frequencies, that are defined by the aforementioned algorithm at thread allocation time, are sent from master PE to DVFS controllers through the network via a specific packet that is illustrated in Figure 11. The Amount Of Voltages and Counter Bit Width (N and M) fields have their bit widths parameterizable, as well as the DVFS Service ID. From now on, we will refer to Counter Bit Width as Counter Resolution. The greater the Counter Resolution, the more accurately clock frequencies are generated at each DVFS domain.

To simplify the process to find a target frequency \( f_{\text{target}} \) using a base frequency \( f_{\text{base}} \) and an ideal target frequency \( f_{\text{ideal}} \), we have used the Counter Bit width as follows:

\[
 f_{\text{target}} = \frac{N}{M} \times f_{\text{base}}
\]

N and M are determined such that \( f_{\text{target}} \) is the smallest number that satisfies

\[
 f_{\text{target}} \geq f_{\text{ideal}}
\]

For instance, if DVFS Counter Resolution = 4, \( f_{\text{base}} = 250 \text{ MHz} \) and \( f_{\text{ideal}} = 100 \text{ MHz} \); \( f_{\text{target}} = 109.375 \text{ MHz} \) through N and M values of 7 and 16 (two in the power of M), respectively.

The Supply Voltage value is set as the largest integer SV that satisfies

\[
 \frac{1}{2SV} \geq \frac{f_{\text{target}}}{f_{\text{base}}}
\]

and

\[
 0 \leq SV \leq \text{AmountOfVoltages} - 1
\]

For instance, when one has \( f_{\text{base}} = 250 \text{ MHz} \) and \( \text{Amount Of Voltages} = 2 \), \( SV \) will be 0 for \( f_{\text{target}} \geq 125 \text{ MHz} \), else, 1.

The IsNoC bit is set as 1 if the packet describes a VF-pair that is intended for a Router. Otherwise, it is set as 0 if the packet describes a VF-pair that is intended for a Bus/Crossbar.

The DVFS controller’s data path can be visualized in Figure 12. DVFS controllers are replicated in the following manner: One per Router in the NoC, plus one for every Bus/Crossbar instantiated. The Credit, Tx and Data input ports of each controller are connected to the local ports of Routers in the NoC. For Routers associated with a PE, this is done directly in the link between the Router’s local port and the PE. For Routers associated with a Bus/Crossbar, this is done in the link between the Router’s local port and the Bus/Crossbar.

Following the proposal of [8], clock signals, for each clock domain, are generated through clock gating of a reference clock signal. In our proposal, this reference clock is gated on a \( N/M \) ratio, where \( N \) is the number of cycles it will be enabled in a window of \( M \) cycles. To accomplish this, a cycle counter is necessary, assuming values between 0 and \( M - 1 \). When this counter’s value is less than \( N \), the clock signal will be enabled, otherwise, it will be disabled.

V. Experimental Evaluation

To demonstrate and quantify the effectiveness of the use of hybrid network topologies and the proposed DVFS mechanism, we have developed a framework that supports RTL simulation of different hybrid topologies and traffic patterns [19].

A. Methodology

A.1 Applications and Workloads - Workloads extracted from relevant real-world multimedia applications are used to stimulate the network topologies to be explored. Figure 13 shows the application graphs used in the experiments. In these graphs, nodes represent threads of an application, while edges and their weights represent the expected Throughput between threads, in MBps. The application characterizations are taken from [20] and [21].

We have used commercial applications to evaluate the communication structures aiming to get results based on real communication scenarios. The applications selected for our experiments are very heterogeneous in terms of throughput. As it can be seen in Figure 13, PIP application does not stress the NoC link as much as the MPEG4 application. For instance, we have a huge stress in the NoC link where the SDRAM2 thread of the MPEG4 application is located. The total throughput caused in the SDRAM2 link is 921 MB/s being almost the maximum bandwidth of a structure link, which is 1Gbyte/second (Bandwidth = 4 bytes*250MHz). However, if such a thread is located in a link where the operating frequency is 150MHz, congestion will be generated since the highest bandwidth is 600MB/s. In contrast, the PIP thread that most stressed a link is the ImpMemA, causing an throughput of 192 MB/s, which produces a lot of room to reduce voltage and frequency in the link. Thus, the heterogeneous workload covers a wide range of communication throughput by variably stressing the structures as the real applications perform, at times it provides congestion and at other times it smoothes traffic.
The application characteristics are exposed in Table I, and the workloads in Table II. For elaborating Workloads from these applications, two properties were taken into account, Throughput Variance and Throughput Demand, which correspond to the Throughput variance and mean of the edges in the application’s characterization graphs, respectively. Three workloads (LL, MM and HH) were elaborated to represent an increasing average amount of each property. A fourth workload (VV) is built to represent a higher variation of both properties within its applications, containing both high and low quantities of them. Workloads are named according to their Throughput Variance and Throughput Demand characteristics (for example, workload LL36 has a (L)ow amount of Throughput Variance and a (L)ow amount of Throughput Demand).

Table I. Emulated applications information

<table>
<thead>
<tr>
<th>Application</th>
<th>PIP</th>
<th>MWD</th>
<th>VOPD</th>
<th>MPEG4</th>
<th>H264,30</th>
<th>H264,60</th>
</tr>
</thead>
<tbody>
<tr>
<td># Threads</td>
<td>5</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>AVG Tp</td>
<td>72</td>
<td>86.15</td>
<td>232.93</td>
<td>266.62</td>
<td>48.28</td>
<td>97.96</td>
</tr>
<tr>
<td>STDDEV Tp</td>
<td>22.63</td>
<td>19.38</td>
<td>160.13</td>
<td>297.25</td>
<td>44.65</td>
<td>92.60</td>
</tr>
<tr>
<td>Tp Variance</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Tp Demand</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Tp Variance #</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Tp Demand #</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Max Tp Thread</td>
<td>192</td>
<td>192</td>
<td>600</td>
<td>921</td>
<td>300</td>
<td>600</td>
</tr>
<tr>
<td>Thread Name</td>
<td>InpMemA</td>
<td>IN</td>
<td>VopRec</td>
<td>SRAM2</td>
<td>YUVGen</td>
<td>YUVGen</td>
</tr>
</tbody>
</table>

Table II. Workloads information

<table>
<thead>
<tr>
<th>Workload</th>
<th>LL</th>
<th>SM</th>
<th>HH</th>
<th>VV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application 1</td>
<td>PIP</td>
<td>MWD</td>
<td>H264,20</td>
<td>MPEG4</td>
</tr>
<tr>
<td>Application 2</td>
<td>H264,60</td>
<td>MPEG4</td>
<td>H264,60</td>
<td></td>
</tr>
<tr>
<td>Application 3</td>
<td>H264,30</td>
<td>VOPD</td>
<td>MPEG4</td>
<td></td>
</tr>
<tr>
<td># Of Threads</td>
<td>35</td>
<td>30</td>
<td>36</td>
<td>35</td>
</tr>
</tbody>
</table>

A.2 Network Topologies - As shown in Table III, four hybrid network topologies were built to evaluate the aforementioned workloads. The process of elaborating network topologies was based on the metrics of Throughput Variance and Throughput Demand. The hybrid topologies were named according to: their Throughput Variance; Throughput Demand characteristics and number of PEs. For example, topology HL36 has a (H)igh amount of Throughput Variance, (L)ow amount of Throughput Demand, and 36 PEs.

Intuitively, it can be understood that the ideal network topology for a workload with a low Throughput Demand would be a Bus, from its inherent low complexity, enabling sufficient performance at low power and area costs. On the opposite end, a Crossbar would be more well-suited for a workload with high Throughput Demand, as it can provide a greater amount of performance than a Bus or NoC.

For workloads with high Throughput Variance, an ideal hybrid topology would consist of several both Buses and Crossbars, since each application on it (or specific threads of applications) would greatly benefit from the heterogeneity of the topology. In such a scenario, threads with low Throughput Demand would be allocated to Buses, while threads with high Throughput Demand would be allocated to Crossbars. Again, this allows for a more precise match of application performance demands and network performance offering, while reducing power and area costs. Complementary, for workloads with low Throughput Variance, a more homogeneous network would be better. As shown in Table III, aiming to accomplish the Throughput Variance and Demand methodology, the LL topology has two buses with eight cores each and one NoC with twenty cores, which provides low variance and low demand. On the other hand, to support high variance and to provide high demand, the HH topology has three crossbars with eight cores each and one NoC with twelve cores. The HL and LH topologies stay between the aforementioned topologies by merging buses, crossbars and NoC.

Considering the process of allocating threads to a given topology, no formal algorithm was employed, but we follow the mentioned principles of affinity between threads and networking elements. For a given topology and workload, Crossbars in the topology are filled with the workload’s threads that show a high Throughput Demand, as taken from the characterization graphs in Figure 13. Similarly, Buses are filled with the threads that show the lowest Throughput Demands, while the remaining threads are allocated to the NoC. Threads that are allocated to the NoC are assigned specific routers such that hop counts and link contention are minimized.
B. Area and Power Results without DVFS

To explore the trade-off between power and area, we have performed logic syntheses of the four hybrid topologies, as well as a 2D mesh Hermes NoC with 36 routers for reference. These syntheses were done in a Multi-Mode Multi-Corner (MMMC) flow with 2 voltage corners, using Cadence Genus with a 45nm PDK. For all scenarios, a buffer size of 4 flits and a Data Width (DW) of 32 bits were used.

Figures 14 and 15 show the significant impact in power and area of the NoC in all Hybrid topologies. The NoC is responsible for between 60% and 85% of power consumption and between 77% and 92% of the area occupied. Furthermore, comparing the LH36 and Hermes36 topologies, 22% of power and 42% of area can be saved by employing a hybrid topology instead of a homogeneous NoC. Such results motivate the employment of a Hybrid network to achieve better tradeoffs between performance, power and area.

C. Comparing Workloads and Network Topologies under DVFS

RTL simulations of each workload and topologies were performed to gather Power, Throughput and Latency results of the four hybrid topologies and Hermes NoC. For these simulations, the parameters of the DVFS controller were set as follows: Counter resolutions assume values of 2, 8 and 14 bits, such that there are 4, 256 and 16384 possible clock frequencies for each resolution, respectively. The amount of supply voltages is set as 2 (corresponding to the 0.9 V and 1.08 V values from synthesis), and each supply voltage value aggregates half of the possible clock frequencies. The maximum and minimum operating frequencies are 250 MHz and 0 MHz, respectively.

The power, throughput and latency results of the simulations for 3 different DVFS controller counter bit width values are presented in Figures 16, 17, 18, normalized to the associated No-DVFS case.

Analyzing the impact of the DVFS aggressiveness over the latency and power, by looking at different colors in the bars, it is clear that more aggressive DVFS (Router-grained (blue bars)) provides higher power savings than the coarser approaches, but it penalizes overall latency. Such behavior is seen in the experiments with Workload LL. In the most evident case, topology LH36 (third graph of first row) and counter resolution of 14, DVFS in Router granularity saves 89% of power, but increases 635% in overall packet latency in comparison to No-DVFS results. The finer the DVFS is, the more gains are shown in power, however higher latency is observed. On average, the Router-, Structure- and Global-grained provide 54%, 48% and 28% power savings and increase latency in 220%, 133% and 138%, respectively.

The aforementioned relationship between power and latency is not observed between throughput and power. Even though power is seen to be greatly reduced with the aggressiveness of the DVFS, the same is not observed in throughput. Such behavior is especially noticed for Workloads LL (graphs of the first row) and MM (graphs of the second row), since in these scenarios the DVFS approach has more opportunities to change VF pairs than the Workload HH (graphs of the third row), where applications are always requesting the highest bandwidth. Throughputs remain roughly the same for LL and MM, while power consumption is significantly decreased, especially on applying Router-grained DVFS. On average, power savings of 80% and 66% are shown in the Workload LL and MM, respectively, when the Router-Grained approach is applied. In all scenarios, DVFS always produces more power savings than throughput losses, on average, 43% of power savings is observed, while throughput is penalized by 12%.

By analyzing the impact of the counter resolution of the DVFS controllers in power savings, one can notice in Workload LL and MM graphs of Figure 16 (first and second row), the higher the counter resolution, the higher power savings is shown. The network frequency more closely matches the ideal frequency when the number of available frequencies is higher. However, the increase in counter resolution implies a Latency cost, even though Throughput remains roughly unaffected. On average, power savings of 36%, 46% and 47% are shown with 2, 8 and 14 counter resolutions with an increase in latency of 117%, 169% and 207% when the Workload LL and MM are considered. As the opportunities for DVFS in the Workload HH (graphs of the third row) are reduced, the increase of counter resolution does not affect power, but increases latency.

Throughput losses are observed in Workloads MM, HH and VV, which would not be a suitable behavior for our DVFS approach since it considers Throughput as a non-violable requirement. In such cases, pessimistic frequency sets are performed by the DVFS since the main goal of the technique is to achieve near-optimal power savings. Throughput is penalized by 12% on average, which is reasonable considering the diversity of simulations performed.

Finally, we have analyzed the efficiency of having DVFS coupled to a Hybrid Topology by comparing it against a Her-
Fig. 16 Normalized power in DVFS experiments

Fig. 17 Normalized throughput in DVFS experiments

Fig. 18 Normalized latency in DVFS experiments
mes NoC coupled to the same DVFS logic. The fifth graph column of Figures 16, 17 and 18 show the results of Hermes NoC considering DVFS in Router- and Global-grained. As it can be noticed, considering workload LL (first row of Figure), all Hybrid topologies outperform the Hermes NoC in terms of latency and power savings for all DVFS counter resolution. For workloads MM and HH, as the traffic is heavier than workload LL, the Hybrid Topology LL36 with counter resolution 14 shows a little higher latency than Hermes NoC, since the goal of such a hybrid structure is to occupy a minimum area by using buses. However, for the remaining scenarios and counter resolution of workload MM, the Hybrid structures outperform Hermes NoC in latency and power savings.

D. Power and Area Costs of Implementing DVFS

The cost of implementing DVFS comes through 4 elements: The DVFS controller itself; the Power switches used to select supply voltages; the Level shifter cells that assure correct logic levels between different power domains; and the Voltage regulator that generates the supply voltages to be selected by each power domain.

The DVFS controller was synthesized under 0.9 V voltage and 250 MHz frequency constraints. The Amount Of Voltages parameter was taken as 2, while the Counter Bit Width parameter was taken as 5. This bit-width value is seen as reasonable, since it allows for a frequency step of 7.8125 MHz when used with an input clock of 250 MHz, making possible fine-grained frequency tuning. Synthesis information for the DVFS controller can be visualized in Table IV.

From Table IV, it can be seen that the DVFS contributes to total power consumption to the order of mW, while the power consumption is to the order of mW (from Figure 15). The same proportion is observed of area, which, in the network topologies being studied, is to the order of hundreds of thousands of µm², while the area of the controller is to the order of hundreds of µm². This makes the addition of DVFS controllers of small impact on the area and power consumption of the whole interconnection network, even at the finest granularity.

In [8], synthesis results are reported for a technology node with a feature size of 45 nm. Conveniently, in this work, synthesis results are also obtained from a 45 nm technology node. The authors report an area cost of 25 µm² for two supply switches, plus additional gate driver circuitry. A voltage drop of less than 0.5% in the power switches was observed, making the power consumption of the switches themselves negligible.

As for level shifters, the X1 strength 0.9V-to-1.08V cell in the PDK used is defined to have an area of 11.628 µm². Considering the worst-case scenario for the number of level shifters, Router-grained DVFS in the Hermes36 topology, each Router will require (32 + 1 + 1) level shifters per port, totaling 5440 level shifting cells required (32 data lines + Tx + Credit, times the number of router ports not at the edge of the NoC, times the amount of routers). Assuming them to be

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Total Power (mW) & Total Area (µm²) \\
\hline
46.788 & 631.023 \\
\hline
\end{tabular}
\caption{DVFS controller synthesis information}
\end{table}

the mentioned cell, this results in a significant 63256.32 µm² area overhead, which corresponds to 9.8% of the cell area of Hermes36.

Finally, [11] argues that costs for an integrated regulator that makes DVFS possible cannot be attributed solely to the interconnection network. Seeing as the supply voltage generated by such a regulator will most likely be used in other modules throughout the system, power and area costs must be split between the additional components that use this additional supply voltage. Since there is no way to estimate possible supply voltage usage by different modules, no reliable values for the cost of a voltage regulator can be determined. Nevertheless, seeing there are 36 PEs serviced by the interconnection network, it can be assumed that the amount of PEs that use the generated voltage will be numerous enough that the costs of a regulator, diluted between them and the interconnect, will be low enough as to make the addition of DVFS worthwhile, as far as a voltage regulator is concerned.

VI. CONCLUSION

The use of hybrid topologies in on-chip interconnection networks is demonstrated to be an effective way of obtaining power and area savings. When compared to a homogeneous NoC with the same number of PEs, gains of up to 22% in power and 42% in area (Topology LH36) can be obtained through a hybrid topology interconnection network. Further power savings can be achieved by the use of DVFS. By discretizing possible supply voltages, local generation of clock signals and software-based centralized decision making, DVFS can be implemented such that area overheads and additional system complexity are minimized while power efficiency is maximized. Experiments with popular video encoding applications show an improvement of up to 89% in power consumption (Topology HL36, Workload LL, Counter Resolution 14), while application performance is not compromised in most, but not all cases. The implementation of fine-grained DVFS has non-trivial area costs, but still presents a favorable outcome as far as the area-power trade-off is concerned.

For future works, additional exploration of hybrid topologies can be done. As in this work, a NoC-centric approach was taken, but many other possibilities are left to be explored. Such opportunities may be as an array of Buses interconnected by a Crossbar, or the use of hierarchical Buses/Crossbars within other Buses/Crossbars. On the DVFS aspect of our proposal, it can be seen that not all Throughput demands were supplied by the network under fine-grained DVFS, making obvious room for improvement in the methods used for computing networking elements’ clock frequencies. Once this is accomplished, a concrete DVFS-aware thread allocation algorithm, based on previous works in [4] and [5] might be developed, to provide optimal task allocation, allowing for minimal power consumption while still meeting application Throughput demands.

\textbf{REFERENCES}
