# Analysis and Mapping for Thermal and Energy Efficiency of 3-D Video Processing on 3-D Multicore Processors

Amit Kumar Singh, *Member, IEEE*, Muhammad Shafique, *Member, IEEE*,
Akash Kumar, *Senior Member, IEEE*, and Jörg Henkel, *Fellow, IEEE*

***Abstract*— Three-dimensional video processing has high computation requirements and multicore processors realized in 3-D integrated circuits (ICs) provide promising high performance computing platforms. However, the conventional approaches to accelerate the computations involved in 3-D video processing do not exploit the high performance potential of 3-D ICs. In this paper, we propose an application-driven methodology that performs efficient mapping of 3-D video applications' components on 3-D multicores to achieve high performance (throughput). The methodology involves an extensive application analysis to exploit the spatial and temporal correlation available in 3-D neighborhood. Afterward, it leverages the correlation and thermal properties of different 3-D views to perform an efficient mapping of 3-D video processing on cores available at different layers of 3-D IC. The goal is to optimize energy consumption and peak temperature while meeting the throughput requirement. Experiments show 76% reduction in communication energy along with reduction in peak temperature when compared with approaches exploiting architecture characteristics only.**

***Index Terms*— 3-D multicore, 3-D video, design-time analysis, interconnect energy, synchronous dataflow, thermal-aware mapping, throughput.**

## I. Introduction

THREE-DIMENSIONAL services are envisaged to play an important role toward enhancing the future of several industries, such as consumer/entertainment, security, medical imaging, and communication. The advancement in 3-D video technologies [1] and emerging users' sensation for true 3-D reality have evolved new application domains, such as 3-D television, 3-D surveillance [2], and 3-D video recording on the next generation mobile devices [3]. Recent devices released for 3-D recording use two views[1] [4], [5]. However, an increase in the number of views is expected for such upcoming devices to fulfill the emerging market needs.

To address the processing challenges with increased number of views in 3-D video encoding, multiview video coding (MVC) [6] standard was devised a couple of years ago. MVC provides up to 50% bitrate reduction (compression) compared with independent coding of different views using the state-of-the-art H.264 video coding standard. This is achieved by exploiting temporal and interview correlation through multiple block-sized motion estimation (ME) and disparity estimation (DE) that in turn significantly increases the computational complexity. The complexity and workload of ME/DE highly depends upon the application specific properties, such as a picture prediction structure,[2] correlation between frames/pictures,[3] and motion/disparity contents in the video sequences.

Several efforts have been made to accelerate the ME and DE computation process in order to achieve high throughput. These efforts use either fast ME/DE algorithms [7]–[9] or hardware acceleration [10], [11]. Although these state-of-the-art ME/DE algorithms and hardware accelerations provide significant computation reduction, they do not exploit full potential of 3-D neighborhood correlation available in spatial and temporal domains. Furthermore, they target 2-D architectures and simply extending them for 3-D multicore architectures to accelerate the computations leads to increased complexity and inefficiency due to significantly different thermal behaviors of 3-D integrated circuits (ICs).

Three-dimensional ICs provide attractive possibilities to implement multicore systems and are regarded as promising future high performance computing platforms. In 3-D ICs, multiple logic layers are stacked vertically, and the layers are connected by through silicon vias (TSVs). Such ICs alleviate performance bottleneck problems incurred due to on-chip interconnects that do not scale in proportion to the process technology [12], and are consid-

Manuscript received May 23, 2015; revised September 1, 2015 and November 6, 2015; accepted December 20, 2015. Date of publication February 4, 2016; date of current version July 22, 2016. This work was supported in part by the Tier 2 Singapore Ministry of Education Academic Research under Grant R-263-000-B33-112, in part by the German Research Foundation within the Cluster of Excellence Center for Advancing Electronics Dresden, in part by the Transregional Collaborative Research Centre Invasive Computing under Grant SFB/TR 89, and in part by the Priority Program Dependable Embedded Systems under Grant SPP 1500.

A. K. Singh is with the Department of Computer Science, University of York, York YO10 5GH, U.K. (e-mail: amit.singh@york.ac.uk).

M. Shafique and J. Henkel are with the Chair for Embedded Systems, Karlsruhe Institute of Technology, Karlsruhe 76131, Germany (e-mail: muhammad.shafique@kit.edu; henkel@kit.edu).

A. Kumar is with the Department of Computer Science, Technische Universität Dresden, Dresden 01062, Germany (e-mail: akash.kumar@tu-dresden.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2016.2517025

[1] Video sequences captured using different cameras.

[2] A prediction structure defines the direction (i.e., previous or future picture) of finding the best match (i.e., the most correlated prediction block) in the search process.

[3] Frames and pictures are interchangeably used for the same thing.

ered as one of the most promising solutions to surmount the interconnect scaling problem [13]. Furthermore, vertical stacking reduces the die size and wire lengths, which results in several advantages, such as reduced production costs, communication delays, and energies [14]. Thus, 3-D multicores achieve higher performance and lower power consumption when compared with the traditional 2-D counterparts [15], [16], and can be considered as potential computing platforms for complex 3-D video processing (encoding).

The communication infrastructure to support communications among the various cores in a 3-D multicore is generally considered as network-on-chip (NoC) due to its scalability and high performance [17]–[19]. On-chip interconnection network consumes a significant portion of the entire chip power [20]. However, this portion might vary between chip architectures and applications running on them. For example, in Intel’s 80-core teraflops processor [20], the network (routers and links) consumes 28% of the total power while operating at 4 GHz, and this ratio increases to 39% at the maximum operating frequency of 5 GHz. Intel’s latest 48-core chip uses advanced power management technique, but the network still consumes 10% of the total chip power [21]. For video processing, the network power might become quite significant as the application components normally have high communication overhead to stream the data among them. Therefore, it is of significant importance to reduce the communication overhead for streaming applications, which will lead to reduced communication time and energy.

In 3-D multicores, vertically aligned cores of different layers are connected using TSVs, which are shorter than the horizontal links [17]. The reduced interconnect distance between vertically aligned cores leads to smaller resistance and capacitance. Furthermore, due to the reduced interconnect distance, vertical interconnects consume much less energy than horizontal links when transmitting the same amount of data [18]. This facilitates allocating heavily communicating tasks on the vertically aligned cores, i.e., in the same core stack to save the communication time and energy. However, TSVs are drilled through the device of each layer by special techniques and are costly to fabricate. In the case of a large number of TSVs, the cost of the 3-D chip will increase. Furthermore, TSV diameters and pitches are quite large as compared with the sizes of regular metal wires. Diameters and pitches are usually ~5–10 $\mu$m and 10–20 $\mu$m, respectively [22]. Thus, the number of TSVs will affect the overall chip areas. Therefore, the number of TSVs needs to be controlled during the chip design, although they provide increased routing and other benefits. Furthermore, placing active tasks within the same stack increases power density, which may result in serious thermal issues as high temperature affects performance, reliability, and lifetime of the system [23]. Therefore, thermal measures are required while accelerating 3-D video processing on a 3-D multicore.

### A. Motivational Example

A motivational example to map a 3-D video with two views (V0 and V1) on a 3-D multicore with three layers is shown in Fig. 1. Each view contains a group of pictures, and spatial (within a video frame), temporal (between frames), and interview (between views) correlations exist in 3-D neighborhood. Furthermore, prediction of some of the frames involves heavy computation. Such frames are characterized as hot frames. For performing thermal and performance aware mapping, compute intensive (hot) components (views, frames) can be placed on layers close to the heat sink (i.e., on the coolest layer) to achieve a good and balanced thermal profile. For example, in Fig. 1, hot frames T0 and T8 of view V0 can be mapped on the cores of the coolest layer Layer0. Moreover, the highly correlated components on adjacent layers (or close to each other) to minimize the communication overhead toward achieving high performance and low energy consumption. For example, correlated frames T1 and T2 of view V0 can be mapped on cores stacked on top of each other and located in the adjacent layers Layer0 and Layer1. However, a straightforward assignment will not lead to good results due to the interview (frame) dependencies. Therefore, there is a need to balance between computation and communication (correlation) induced thermal effects.

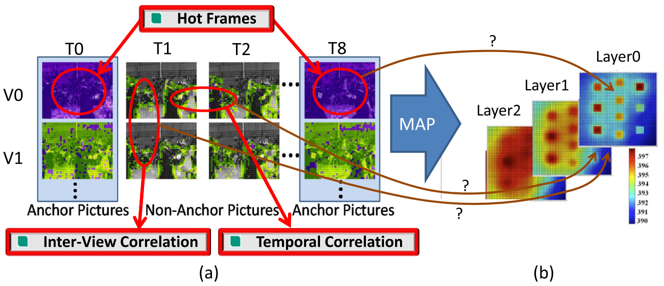


Fig. 1. Exploiting 3-D neighborhood characteristics to map 3-D video on 3-D multicore. (a) 3-D video with two views. (b) Multiple layers of 3-D multicore.

In short, there is a need to devise a methodology that should first analyze the 3-D video processing to identify certain characteristics from 3-D multicore point of view, and then perform mapping by taking application and platform characteristics into account while optimizing for energy consumption and peak temperature.

### B. Our Novel Contributions

This paper addresses shortcomings of existing approaches to perform 3-D video processing on 3-D multicores by providing the following contributions.

1) An analysis strategy to analyze 3-D video processing flow in order to extract the characteristics, such as hot/cold and correlated views/frames.
2) A mapping strategy to map 3-D video processing on 3-D multicore by jointly taking the application and platform characteristics into account toward achieving high performance, thermal balance, and energy savings.

### C. Open Source Contribution

Deriving throughput-constrained synchronous dataflow graph (SDFG) [24] representation of 3-D video processing in order to facilitate easier analysis and mapping on 3-D multicore. This also enables open-sourcing of 3-D video SDFG. We will make it available online for the community for future research and fair comparisons.

In analysis, the tasks of transformed 3-D video as SDFG having long and short computation times are identified as hot and cold tasks (frames), respectively, and (highly) communicating tasks are identified as (highly) correlated tasks (frames). Dependency between tasks (frames) of different views defines correlation between the views. The mapping strategy systematically maps hot, cold, and correlated frames on 3-D multicore architecture while satisfying the throughput requirement. To the best of our knowledge, this is the first work that addresses mapping of throughput-constrained 3-D video processing on 3-D multicore to jointly exploit the characteristics of 3-D video and 3-D multicore.

The remainder of this paper is organized as follows. Section II reviews the literature in the direction of 3-D video processing acceleration and application mapping on 3-D multicores. Section III introduces the system model and problem definition. Section IV presents the proposed mapping methodology. The experimental results to evaluate our methodology are presented in Section V. Section VI concludes this paper.

## II. Related Work

The state-of-the-art efforts to speed up ME/DE computations in 3-D video processing employ fast algorithms or hardware accelerations. The fast algorithms in [7] and [8] employ variable search range based on disparity maps and cameras geometry, respectively. In [9] and [25], a fast prediction (ME or DE) based on blocks motion intensity and complete DE is proposed. The view-temporal correlation and interview correlation have been exploited in [26] and [27], respectively, in order to reduce the computational complexity. In [28], algorithm and architecture for disparity estimation with minicensus adaptive support are proposed. The hardware designs of ME/DE are also proposed [10], [11]. Although the state-of-the-art ME/DE algorithms and hardware accelerations provide significant computation reduction, they do not exploit full potential of 3-D neighborhood correlation available in spatial and temporal domains. Furthermore, they target 2-D architectures and several thermal optimization approaches exit for them [29], [30], but simply extending them for 3-D multicore architectures leads to increased complexity and inefficiency due to significantly different thermal behaviors of 3-D ICs.

Thermal-aware application mapping and scheduling on 3-D multicores are a well-studied topic [14], [23], [31]–[35]. These approaches perform optimizations at design-time or run-time while trying to minimize hotspots and thermal gradients (spatial, temporal, or both).

Run-time approaches generally try to measure or estimate the current temperature distribution in the chip, and take actions based on that in order to minimize hotspots and thermal gradients (spatial, temporal, or both). Zhu *et al.* [36] exploit workload power characteristics and processor core thermal characteristics for efficient thermal management. Coskun *et al.* [32] reviewed several dynamic mechanisms, such as temperature-triggered dynamic voltage/frequency scaling (DVFS), clock gating, and hot task migration, and proposed a run-time task assignment algorithm that takes the thermal history of cores into account. In [37] and [38], the concept of thermal herding has been used, where the most frequently switched activity or hot jobs are assigned to the cores close to the heat sink and cool jobs to the cores far from the heat sink. A thermal-aware operating system level scheduler for 3-D multicores is proposed in [23]. Kang *et al.* [39] reviewed the work of [23] and introduced peak power and temperature constraints. These methods share the goal of minimizing the peak temperature and thermal gradients without sacrificing performance too much. However, the effect of intertask communication is not considered. Since NoC can dissipate a substantial part of the power budget, which depends upon the network traffic [33], interconnect utilization (energy) should also be considered, which has not been considered in most of the aforementioned works.

Design-time mapping approaches aim at finding a thermal-aware mapping by using a model of the physical chip, or by using general knowledge about the thermal behavior of 3-D ICs. In [33], both temperature and communication load are considered, and a genetic algorithm is used to generate static mappings. The design-time mechanisms considering throughput constraint are reported in [34] and [35], but they cannot provide efficient mapping solutions for 3-D video processing as application characteristics (e.g., hot/cold and correlated frames) cannot be exploited. To summarize, existing mapping approaches perform either application aware mapping by exploiting application characteristics or platform-aware mapping by exploiting platform characteristics.

In contrast to the above strategies, our approach performs the thermal-aware mapping of throughput-constraint 3-D video processing on 3-D multicores by exploiting both 3-D video (application) and 3-D architecture (platform) characteristics. In addition, our approach considers the effect of TSVs on temperature distribution and power dissipation, and minimizes the communication energy. In the case of multiple applications to be mapped and executed concurrently while sharing the system resources, all the tasks can be considered in the mapping process. However, this will need to consider all the possible use cases (scenarios), where each use case represents a set of concurrently running applications. The number of use cases increases exponentially with the number of applications. Furthermore, for each use case, composability analysis needs to be employed to ensure that near optimal mapping has been achieved for each application in order to satisfy the throughput constraints. To avoid evaluation for a huge number of use cases and their composibility analysis, the applications can be mapped and executed one after another without sharing resources. Furthermore, in case, 3-D multicore platform is complex, i.e., contains a large number of cores; multiple applications can be mapped and executed concurrently. Toward this, a set of cores can be reserved for each application at design-time, so that different applications can be mapped and executed into disjoint regions.

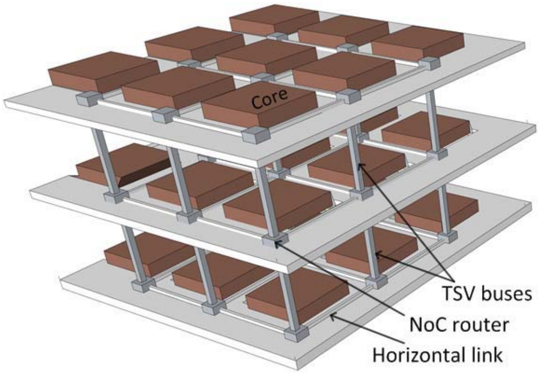


Fig. 2. Example 3-D multicore architecture.

## III. System Model and Problem Formulation

### A. Three-dimensional Multicore Architecture and 3-D IC Model

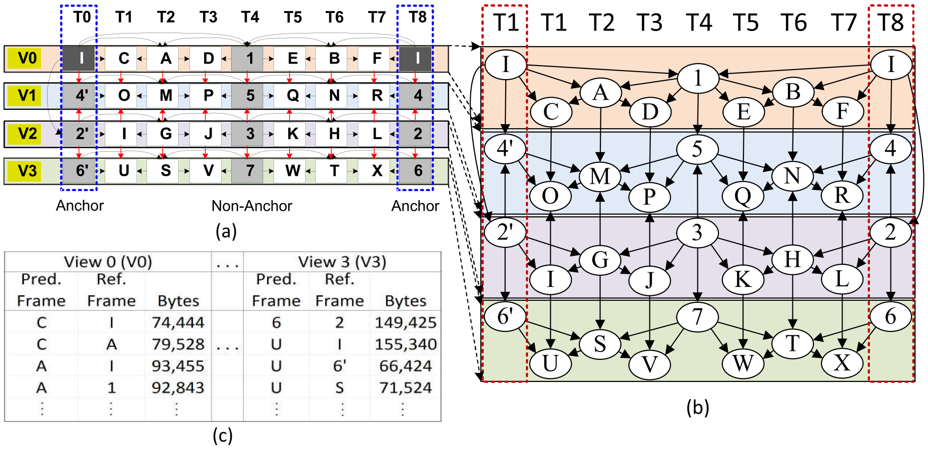
The 3-D multicore is modeled as a regular 3-D mesh of homogeneous cores connected by a NoC, as shown in Fig. 2. For the NoC, similar to [14], a hybrid NoC-Bus design is considered, which consists of a regular NoC in the horizontal plane and a multidrop shared bus (TSVs) to connect the cores within the same stack. Thus, one vertical pillar of TSVs is used for a set of routers that are aligned vertically, and cores within the same stack are accessed in a single hop [40]. TSVs are of shorter length than that of the horizontal links (tens versus thousands of micrometers [14]), and thus, they often provide faster and more energy efficient communication than the horizontal links [12]–[15], [17], [18]. For communicating among vertically aligned cores, the same number of cycles are required (accessed in a single hop [40]) as they are connected by the multidrop shared bus. In contrast, communication among cores situated in the horizontal plane requires a different number of cycles depending upon the hop distance between cores [41]. The core may contain a processing element, for example, ARM processor, and local memory (M). The architecture (platform) is represented as a directed graph PG $= (C, V)$, where $C$ is the set of cores and $V$ represents the connections among the cores. Each core has active power $pa$ and idle power $pi$.

For 3-D IC, a 3-D grid model available in the HotSpot thermal simulator has been employed [42]. The application tasks are executed on cores, and the execution is tracked at core level. The power dissipated in each core is distributed over the blocks (e.g., processor, memory, and router) based on an intracore power distribution. An example distribution case for a core executing a high instruction level parallelism task with low memory traffic is given as 80% power dissipated in the processor block, 10% in the router block, and 10% in the memory. We also have considered similar fine-grained power distribution. This way we can achieve power dissipation in every block. Furthermore, within each considered core, the processor block occupies a significant portion of the total area and dissipates maximum portion of the total core power as mentioned above. Thus, the processor block has maximum power density in terms of power value per area, and its temperature determines the peak temperature. To achieve more fine-grained power distribution, power consumption in different parts of the processor, e.g., registers, arithmetic logic unit, and so on, can also be considered, but this is orthogonal to our focus, which is at core level. To consider the thermal effect of TSVs, their size, position, and material properties are specified in the 3-D IC model [42]. The HotSpot simulator is extended to take TSVs into account, where thermal properties (conductance and heat capacity) of grid cells containing TSV material are changed based on a grid cell volume occupied by TSV material.

### B. 3-D Video Application Model

The MVC prediction structure used to perform 3-D video processing is employed from [11] and shown in Fig. 3(a). The structure is based on four views ($V_0$–$V_3$). MVC uses ME and DE tools to eliminate the temporal and view redundancies between frames, respectively. In Fig. 3(a), $I$ frames are intrapredicted frames (i.e., no ME/DE is used), some frames use unidirectional prediction or estimation [e.g., $2'$, 2, $6'$, and 6 as shown in Fig. 3(a)], and the rest of the frames use bidirectional prediction with reference frames in at least two directions. The arrows represent prediction directions, and frames at the tail side are the reference frames to the frames at arrowheads. To facilitate for access points, the video sequence is segmented in groups of pictures (GOPs), where frames at borders are known as anchor frames that are encoded with no reference to the previous GOP and others are known as nonanchor frames.

The MVC prediction structure has been derived to equivalent SDFG, as shown in Fig. 3(b). The SDFG model is represented as a directed graph AG $= (T, E)$, where $T$ is the set of nodes modeling tasks of the application and $E$ is the set of directed edges modeling dependencies among the tasks. The nodes of an SDFG are also referred to as actors. Each actor represents a frame in the corresponding MVC structure. The execution time of actors (equivalent to ME/DE prediction overheads of frames) and required communication parameters for edges (amount of data required for ME/DE predictions as the number of tokens and their size $TokSize[edge]$) are set by analyzing the execution behavior of MVC [Fig. 3(a)] toward achieving the same execution behavior of equivalent SDFG model [Fig. 3(b)]. Some reference data for two views (View0 and View3) of *Ballroom* video sequence is shown in Fig. 3(c). For different frames (actors) to be predicted (Pred.) in a view, one or multiple frames from the same or other views are used as reference (Ref.) frames. For example, frame $C$ uses frames $I$ and $A$ as reference frames. The amount of transferred data (bytes) required from various frames to predict a frame is provided in the last column. These data values determine the volume of data on the edges and computation time of an actor. The shown data values are for the worst case prediction (involving maximum prediction to encounter fluctuations in the data), and using computation times according to the same helps to model the worst case behavior of the 3-D video processing. The 3-D video processing is also characterized by throughput constraint $\Gamma$, and the same has been incorporated in the SDFG model.



| View 0 (V0) | | | . . . | View 3 (V3) | | |
|---|---|---|---|---|---|---|
| Pred. Frame | Ref. Frame | Bytes | | Pred. Frame | Ref. Frame | Bytes |
| C | I | 74,444 | | 6 | 2 | 149,425 |
| C | A | 79,528 | . . . | U | I | 155,340 |
| A | I | 93,455 | | U | 6' | 66,424 |
| A | 1 | 92,843 | | U | S | 71,524 |
| ⋮ | ⋮ | ⋮ | | ⋮ | ⋮ | ⋮ |

Fig. 3. Three-dimensional video processing with four views and equivalent SDF graph model. (a) MVC prediction structure for 3-D video processing. (b) Equivalent dataflow graph model. (c) Prediction details of *Ballroom* video.

### C. Energy Consumption Model

Communication energy between the communicating cores $c_i$ and $c_j$ (required to predict the current frame by reference frames) depends upon data volume $\text{data}(c_i, c_j)$ and energy required to transfer one bit of data $E_{\text{bit}}(c_i, c_j)$ between the cores. $E_{\text{bit}}(c_i, c_j)$ depends upon the energy required for the horizontal and vertical links traversal and the energy consumed in routers between cores $c_i$ and $c_j$, and is computed as follows:

$$\begin{aligned} E_{\text{bit}}(c_i, c_j) = {} & \left(E_{\text{bit}}^{\text{horizontal}} \times \text{horizontal}_{\text{hops}}(c_i, c_j)\right) \\ & + \left(E_{\text{bit}}^{\text{vertical}} \times \text{vertical}_{\text{hops}}(c_i, c_j)\right) \\ & + \left(E_{\text{bit}}^{\text{router}} \times \text{nrRouters}(c_i, c_j)\right). \end{aligned} \quad (1)$$

In (1), $E_{\text{bit}}^{\text{horizontal}}$ and $E_{\text{bit}}^{\text{vertical}}$ are the energy required to transfer one bit per hop in the horizontal and vertical directions, respectively. $\text{horizontal}_{\text{hops}}(c_i, c_j)$ and $\text{vertical}_{\text{hops}}(c_i, c_j)$ are the number of hops between cores $c_i$ and $c_j$ in the horizontal and vertical directions, respectively. $E_{\text{bit}}^{\text{router}}$ is energy consumed in a router and $\text{nrRouters}(c_i, c_j)$ is the number of routers between cores $c_i$ and $c_j$.

The values of $E_{\text{bit}}^{\text{horizontal}}$, $E_{\text{bit}}^{\text{vertical}}$, and $E_{\text{bit}}^{\text{router}}$ are derived for the 90-nm process technology node (detailed in Section V-A). The router is composed of input first input first output (FIFO) buffers, a fully connected crossbar, and an arbiter. At the input, a three-place FIFO buffer is used. To aggregate the inputs to the outputs, a 6 × 6 full connected crossbar (one extra port to connect to the vertical multidrop shared bus) with multiplexers is used. The arbitration of the router occurs at granularity of words and the routing follows source routing, i.e., path information from source to destination is contained in the header. The details of the router design are available in [43]. The derived value of $E_{\text{bit}}^{\text{router}}$ takes such router structure into account.

The communication energy $E_{\text{comm}}$ is estimated by summing over all communicating task pairs (edges)

$$E_{\text{comm}} = \sum_{\forall \text{comm}-\text{cores}} \text{data}(c_i, c_j) \times E_{\text{bit}}(c_i, c_j). \quad (2)$$

Computation energy required to process all the actors (ME/DE computations) is estimated as follows:

$$E_{\text{comp}} = \sum_{\forall a \in T} a_{\text{ExecTime}} \times pa \quad (3)$$

where $a_{\text{ExecTime}}$ is execution time of actor $a$, and $pa$ is the active power dissipation of core executing actor $a$.

Total energy consumption is measured as sum of $E_{\text{comp}}$ and $E_{\text{comm}}$. The static energy consumption depends upon the static (leakage) power consumption of the cores, which is assumed as a fixed offset. In this paper, we purposely do not account for the power-gating to stay orthogonal to other low-power techniques; therefore, the leakage power will stay constant throughout all of our experiments. Thus, to purely show the impact of the proposed techniques, the results only show the dynamic power consumption. Please note that, any state-of-the-art power-gating technique can be employed in our architecture after the mapping decisions are taken, i.e., power-gating the idle cores. Furthermore, since we consider homogeneous architecture, the computation performed in any part of the architecture will consume almost the same energy. Therefore, the importance of the communication energy becomes the primary focus for optimization/reduction.

### D. Mapping Problem: Three-Dimensional Video Processing on 3-D Multicore

Given SDFG model of a video sequence $\text{AG} = (T, E)$ with throughput constraint $\Gamma$ and 3-D multicore architecture $\text{PG} = (C, V)$.

Find efficient actors to cores mapping to simultaneously optimize for peak temperature (PeakTemp) and communication (interconnect) energy consumption ($E_{\text{comm}}$)

$$\text{PeakTemp} \times E_{\text{comm}} \quad (4)$$

$$\text{s.t. } \tau \leq \Gamma \quad (5)$$

where $\tau$ is the through obtained as a result of the actors to cores mapping and $\Gamma$ is the throughput constraint.

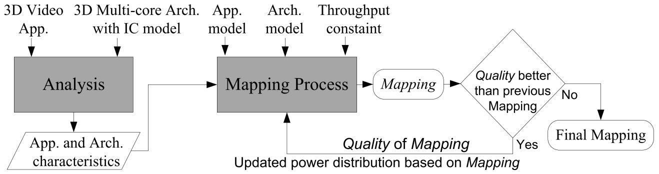

3D Video App.
3D Multi-core Arch. with IC model
App. model
Arch. model
Throughput constaint
Analysis
Mapping Process
Mapping
Quality better than previous Mapping
No
Yes
Final Mapping
App. and Arch. characteristics
Quality of Mapping
Updated power distribution based on Mapping

Fig. 4. Offline analysis and mapping process.

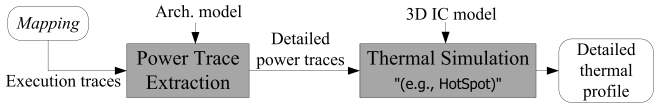


Fig. 5. Temperature estimation flow.

**Algorithm 1** View Level Application Analysis

```
Input: SDFG model of 3D video.
Output: Criticality of views and their actors, correlation between
        views.
for each view $v$ in 3D video do
    weight[$v$] = $\sum_{\forall actors \in v}$ ExecTime[actor];
    criticality[$v$] = weight[$v$]/nrActors;
    for each actor $a$ in $v$ do
        criticality[$v$][$a$] = ExecTime[$a$];
    end
    Find connected views conn_views with $v$;
    for each view $v_{conn}$ in conn_views do
        correlation[$v$][$v_{conn}$] = $\sum_{\forall edges \in v \rightarrow v_{conn}}$ TokSize[edge];
    end
end
```

For a mapping, throughput computation and temperature estimation are in the orders of several milliseconds and seconds, respectively. Therefore, evaluation of all the possible mappings to identify the best mapping (in terms of peak temperature and energy consumption while satisfying the throughput constraint) is expected to take several days. To overcome the evaluation time bottleneck, heuristic-based approaches relying on various cost parameters pertaining to application and architecture characteristics need to be applied to identify an efficient mapping rapidly. However, the identification and extraction of the exact required application and architecture characteristics are challenging specially for large problems, such as 3-D video. We employ the required cost function to the mapping process that considers the desired optimization parameters pertaining to application and architecture characteristics.

## IV. Proposed Mapping Methodology

An overview of our offline mapping identification and temperature estimation flow is shown in Figs. 4 and 5, respectively. The mapping flow first performs offline analysis of the application and 3-D architecture to identify their characteristics that are used to identify efficient thermal-aware mapping (mapping process). The offline analysis considers worst case computation and communication requirements (as described in Section III-B) of 3-D video coding prediction structures for different test video sequences provided by the standardization committee. Such consideration facilitates the worst case online behavior modeling of the 3-D video processing. For the identified mapping for a video sequence, a thermal analysis is performed (temperature measurement) to identify the temperature distribution across different components of the 3-D multicore architecture. In order to perform online video processing for a video sequence, its offline identified mapping is used to configure the platform, and then, the processing starts. The details of offline analysis, mapping process, temperature measurement, and online processing steps are described subsequently.

### *A. Offline Application and Architecture Analysis*

*1) Application Analysis:* The 3-D video application analysis is performed to identify hot, cold, and correlated views and frames (actors). Without loss of generality, let us consider an example case, where the 3-D video processing application contains four views and each view contains nine actors (ME/DE computations) (Fig. 3). In general, the same objects in a 3-D scene are typically present in different views, and motion perceived in one view is directly related to that of the neighboring views [26]. Moreover, the disparity of one given object perceived in two cameras (views) remains the same at various time instances when just translational motion occurs [8], [26], [27]. These observations indicate that there is a high correlation in the 3-D neighborhood that can be exploited during the ME/DE computations. These observations are exploited to set the ME/DE (actor) computation overheads and communication overheads (correlation) between actors. The frames encoded with high effort are referred to as hot frames (actors) and have high computation time, whereas cold actors have low computation time. High communication overhead between actors reflects high correlation between them and vice versa.

We propose two approaches to extract the view and frame level application characteristics to be used by the mapping process.

*a) View level analysis:* The view level analysis (VLA) approach is presented in Algorithm 1. The algorithm takes the application model as input and provides criticalities of views and correlation between them. The criticality of actors within a view is also provided. For each view in a 3-D video, first, view weight (weight[$v$]) is computed, which determines hotness of the view. A view having high weight is considered as a hot view. Then, the view criticality (criticality[$v$]) is computed by dividing the view weight by the number of actors in the view. The criticality of an actor within a view is determined by its computation overhead (execution time). Thereafter, correlation between views is calculated by adding the token sizes (TokSize) of each edge (representing data volume) present between the views. The TokSize is extracted from the application model.

*b) Frame level analysis:* The frame level analysis (FLA) is performed in the similar way as that of VLA, but criticalities and correlations are identified at the frame levels. For the example 3-D video in Fig. 3, there are nine frames in total ($T_0$–$T_8$), and each frame contains four actors (e.g., C, O, I, and U in frame $T_1$). In this analysis, similar steps as

**Algorithm 2** Architecture Analysis Algorithm

```
Input: 3D IC model, Multi-core architecture model
        $PG = (C, V)$, total chip power $P \in \mathbb{R}$, max. # of iterations
        $Iter_{max} \in \mathbb{N}$, terminating condition $\delta \in \mathbb{R}$.
Output: Power ratios of cores ($R_c$ for each core $c$, $c \in C$).
Initialize power ratio for each core as $R_c = 1/N_{cores}$;
$iter = 0$, $Temp_{max} = 500$;
repeat
    $Temp_{prev\_max} = Temp_{max}$;
    Generate power traces for all blocks of each core;
    Simulate steady state temperature distribution.;
    Find peak temperature in each core $Temp_{peak,c}$;
    Find average $Temp_{avg}$ and max. $Temp_{max}$ chip temperature ;
    for each core $c \in C$ do
        $d = (Temp_{peak,c} - Temp_{avg})/Temp_{avg}$;
        if $Temp_{peak,c} > Temp_{avg}$ then
            $R_c = R_c * (1.0 - (\gamma * d)))$; //decrease power ratio
        else
            $R_c = R_c * (1.0 + (\gamma * d)))$; //increase power ratio
        end
    end
    Renormalize power ratios for each core $R_c$;
    $iter++$;
until $(Temp_{prev\_max} - Temp_{max}) \geq \delta$ AND $iter \leq Iter_{max}$;
```

**Algorithm 3** Thermal-Aware Mapping Algorithm

```
Input: $AG$, $PG$, $\Gamma$, Characteristics of $AG$ and $PG$.
Output: Final mapping $FM$.
Quality of previous mapping $Q_{PM}$ = 0;
//Mapping Process
Find criticalities of actors $\in AG$ by using $AG$ characteristics;
Sort all actors $\in AG$ in descending order of criticality;
for each sorted actor $a \in AG$ do
    Find cost of each core $c \in PG$ by using $AG$ & $PG$
    characteristics, as $cost(a, c)$ (see Eqn. 8);
    Sort all cores $\in PG$ in ascending order of $cost$;
    for each sorted core $c \in PG$ do
        if actor $a$ can be bound to core $c$ then
            Assign actor $a$ to core $c$ and incoming/outgoing edges
            of $a$ to connections to construct Mapping $M$;
            break;
        end
    end
end
Compute $\tau$ and quality of $M$ as $Q_M$ (by Eq. 4) ;
if $\tau < \Gamma$ then
    if $Q_M > Q_{PM}$ then
        $Q_{PM} = Q_M$;
        Repeat the Mapping Process with updated power
        distributions on cores based on $M$;
    else
        $FM$ = $M$; break;
    end
end
```

in Algorithm 1 are adopted, and frame level processing by considering frames and connected frames is performed to calculate criticalities of frames, their actors and correlation between frames.

The VLA and FLA approaches extract application characteristics at two different granularities. Based on suitable scenarios, one can provide better characteristics than other to facilitate for efficient mapping. The effect of utilizing such various characteristics in the mapping process is demonstrated in Section V.

The complexity of the analysis depends upon the number of operations to be performed for the criticality and correlation calculations. For $n$ actors, and $e_v$ and $e_f$ edges between views and frames, respectively, the complexity of VLA and FLA is O($n \times e_v$) and O($n \times e_f$), respectively. Since the number of edges between frames is higher in FLA than that of VLA (Fig. 3), FLA has slightly higher complexity.

*2) Architecture Analysis:* The 3-D architectures have been analyzed to observe their thermal and power dissipation characteristics [23], [32], [35]. In 3-D architecture, if the same amount of power has to be dissipated by all the cores on different layers, the cores that close to heat sink will dissipate faster than others, and thus, high temperature gradient and peak temperature will be achieved. The architecture analysis approach presented in Algorithm 2 is used to extract the platform thermal characteristics as the power distribution among the cores, such that peak temperature and temperature gradients are minimized. The algorithm finds the power ratios of cores based on a steady-state temperature distribution resulting from earlier power distribution. The power ratio of a core $R_c$ is defined as the ratio of power dissipated in the core $c$ and total chip power $P$. The approach decreases the power ratios of cores having peak temperature greater than the average temperature and vice versa until temperature difference among cores is reached to a very low value. The increment and decrement are done in small steps by setting a low integer value of the adaptation constant $\gamma$. In agreement to general observations [23], [32], [35], the cores that close to the heat sink get higher power ratios. The power ratio of every core is passed as the architecture characteristic to the mapping process.

The complexity of architecture analysis (Algorithm 2) depends on steady-state temperature simulation and number of iterations. The simulation time depends on the spatial resolution and the number of layers, and it takes ~2 min on a 1.70-GHz Intel i5 CPU for an IC with three layers and resolution of 32 × 32. The algorithm usually converges in five to ten iterations, and thus, the whole analysis takes up to 20 min.

### *B. Offline Mapping Computation*

The steps followed by the proposed mapping algorithm are described in Algorithm 3. The algorithm incorporates thermal awareness to find an efficient final mapping by exploiting (thermal) characteristics of application and architecture, and updated power distributions based on intermediate mapping. First, the criticalities of all actors are computed, and the actors are sorted in the descending order of their criticality. The criticality of an actor cric[a] is calculated by exploiting application characteristics extracted by VLA or FLA (described earlier) as follows:

$$\mathrm{cric}[a]_{\mathrm{VLA}} = k_1 * \mathrm{criticality}[v] + k_2 * \mathrm{correlation}[v][v_{\mathrm{conn}}] \tag{6}$$

$$\mathrm{cric}[a]_{\mathrm{FLA}} = k_1 * \mathrm{criticality}[f] + k_2 * \mathrm{correlation}[f][f_{\mathrm{conn}}] \tag{7}$$

where $\upsilon$ and $f$ are, respectively, the view and frame containing actor $a$. $\upsilon_{\text{conn}}$ and $f_{\text{conn}}$ are the connected view and frame from $\upsilon$ and $f$, respectively. It should be noted that either $\text{cric}[a]_{\text{VLA}}$ or $\text{cric}[a]_{\text{FLA}}$ is used based on the employed application analysis approach VLA or FLA.

Sorting of actors as described earlier helps to handle their mapping systematically, for example, first actors from hot views and then from correlated views by giving a higher value to $k1$ than $k2$, and vice versa. Then, by following the steps in Algorithm 3, sorted actors are assigned one by one on the cores that can support them and incur minimum assignment cost computed as follows:

$$\text{cost}(a, c) = c_1 * \text{LB}(a, c) + c_2 * \text{PCE}(a, c) + c_3 * \text{ACE}(a, c) \tag{8}$$

where $\text{LB}(a, c)$, $\text{PCE}(a, c)$, and $\text{ACE}(a, c)$ represent the normalized processor load, cost for platform characteristic exploitation (PCE), and cost for application characteristic exploitation (ACE) when actor $a$ is bound to core $c$, and $c_1$, $c_2$, and $c_3$ are the weights given to different optimization criteria.

The ACE uses the view or frame level characteristics exploited from VLA/FLA, where average latency of all edges to/from $a$ is minimized by mapping connected actors close to each other in order to exploit interview (frame) correlations. This results in reduced communication overhead/energy. For view and frame level exploitation, ACE is referred to as AVCE and AFCE, respectively.

The PCE uses cost for the power ratio balancing of core $c$ [$\text{PRTB}(a, c)$] and cost for the power ratio balancing of stack containing $c$ [$\text{PRSB}(a, c)$], and is obtained by adding both the costs. $\text{PRTB}(a, c)$ is computed by dividing the estimated power ratio of core $c$ (when assigning $a$ to $c$) by the power ratio of $c$ ($R_c$) suggested by the architecture analysis. Similarly, $\text{PRSB}(a, c)$ is computed by dividing the estimated power ratio of stack $s$ (containing $c$) when binding $a$ to $c$ by the power ratio of stack $R_s$ that is computed by summing up the power ratios of all the cores in $s$. A core stack $s$ consists of a set of cores having the same horizontal position in different layers. Since strong thermal correlation exists between vertically adjacent cores [14], [23], [35], it might be beneficial to consider power ratios of stacks (PRSB) to distinguish in the incurred costs when deviations from original power distribution in the vertical and horizontal directions are the same. This helps to achieve better results from thermal perspective.

The mapping algorithm assigns all actors ($\in$AG) to cores ($\in$PG) and connections to memories inside cores or interconnect links. The mapping process repeats itself to identify a better quality of throughput satisfying mapping by considering updated power distributions on cores based on the current mapping (Fig. 4). This iterative refinement process is expected to lead to a high-quality mapping. The quality of the mapping, $Q_M$, is computed by employing (4), which requires peak temperature and communication energy, and computes the quality as the product of peak temperature and communication energy. If the product value is low, then the mapping is considered to have a good quality. This quality is used to compare the mapping with the previous evaluated mapping ($Q_{\text{PM}}$) in the iterative refinement process. The throughput, energy consumption, and temperature estimations for a mapping are done as follows.

**Algorithm 4** Online Video Processing

```
Input: Video Sequence, $PG$, Offline computed mappings.
Output: mapping to start online processing.
Select the mapping from offline computed mappings for the video sequence;
Configure platform $PG$ based on mapping;
Start video processing;
```

The throughput computation is performed by employing the technique of [44]. However, any throughput computation technique can be employed, which is orthogonal to our focus. In [44], the throughput for a mapping is computed by taking the resource allocations into account. First, a static-order schedule for each core is constructed that orders the execution of bound actors. A list-scheduler is used to construct the static-order schedules for all the cores at once. Then, all the binding and scheduling decisions are modeled in a graph called binding-aware SDFG. Finally, self-timed state-space exploration of the binding-aware SDFG is performed to compute the throughput, which is the inverse of the long-term period, i.e., the average time needed for one iteration of the application. In doing so, the mathematical model used in [44] takes computation, communication, latency for data arrival, and jitter into account. This indicates that throughput depends on the mapping. The dependence of throughput on the mapping has also been well studied in [41].

The energy consumption is computed by employing the approach described in Section III-C. The temperature estimation flow is shown in Fig. 5. In order to get the detailed temperature profile resulting from a mapping, first, the execution traces are generated. The execution trace of each core represents its active and idle time intervals. For active and idle intervals, the core is assumed to consume active and idle powers, respectively. The power traces are used to simulate the temperature with the modified HotSpot thermal simulator (further details in Section V).

### C. Online Video Processing

To perform online processing for a required video sequence, first, the actors of the application are loaded (configured) onto the platform resources based on the offline computed final mapping for the video sequence and then real processing starts. The online video processing for a video sequence follows Algorithm 4. It selects the offline computed mapping for the video sequence, and the platform is configured based on the same in order to start the video processing. For a video sequence, the online mapping is performed once at the application startup and has a small overhead (quantitative description in Section V).

## V. Performance Evaluation

### A. Experimental Setup

The proposed thermal-aware mapping methodology has been implemented as an extension of the publicly available

TABLE I

3-D IC PARAMETERS

| Parameter | Value |
|---|---|
| Technology node [nano $m$] | 90 |
| Each core size [$mm \times mm$] | 2 × 2 |
| TSV diameter [$\mu m$] | 10 |
| TSV pitch [$\mu m$] | 20 |
| Horizontal hop delay [time-units] | 2 |
| Vertical hop delay [time-units] | 1 |
| Bottom layer thickness [$\mu m$] | 200 |
| Non-bottom layer thickness [$\mu m$] | 50 |
| TIM layer thickness [$\mu m$] | 10 |
| Heatsink side/thickness [$mm$] | 14 × 14 × 10 |

TABLE II

THERMAL SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Silicon thermal conductance [$W/(m \cdot K)$] | 150 |
| Silicon specific heat [$J/(m^3 \cdot K)$] | $1.75 \cdot 10^6$ |
| TIM thermal conductance [$W/(m \cdot K)$] | 4 |
| TIM specific heat [$J/(m^3 \cdot K)$] | $4 \cdot 10^6$ |
| TSV thermal conductance [$W/(m \cdot K)$] | 300 |
| TSV specific heat [$J/(m^3 \cdot K)$] | $3.5 \cdot 10^6$ |
| Convection resistance to ambient [$K/W$] | 3.0 |
| Heatsink thermal conductance [$W/(m \cdot K)$] | 400 |
| Heatsink specific heat [$J/(m^3 \cdot K)$] | $3.55 \cdot 10^6$ |
| HotSpot grid resolution | 32 × 32 |
| Temporal resolution [$\mu s$] | 10 |
| Ambient temperature [$K$] | 300 |

SDF$^3$ tool set [45]. As a benchmark to evaluate the quality of the methodology, models of four evaluated video sequences *Ballroom*, *Vassar*, *Crowd*, and *Kendo* (recommended by joint video team in multiview test conditions [11]) have been considered.

Target 3-D multicore platforms contain a different number of cores, where active power of each core is set to 1.5 W and idle power is 10% of the active power [46]. The size of each core is set to 2 ×2 mm$^2$, which is derived by extrapolating the size of the core in the 90-nm technology node. The heatsink is connected (via a heat spreader) to the bottom layer and has a thickness of 200 $\mu$m, and the other active (power dissipating) layers are assumed to be thinned to 50 $\mu$m for better heat conductivity [14], [23]. Between two active layers, a 10-$\mu$m thin layer containing thermal interface material is used. For vertical communication, each TSV bundle contains 8 × 9 TSVs. In our considered core size (2 × 2 mm$^2$), 1 × 1 mm$^2$ is allocated for the router and rest of the area for processor and memory. The area allocated for router (1×1 mm$^2$) is used to place the 8×9 TVSs and is sufficient to accommodate them. Some other important physical properties of the 3-D IC model are summarized in Table I.

Temperature is estimated by employing extended HotSpot thermal simulation tool [42]. To estimate temperature resulting from a mapping, an execution trace of 0.5 s is generated. The execution patterns are periodic with a period much shorter than 0.5 s, and thus, longer simulations become obsolete. Power traces for every block are derived from the execution trace and the architecture specification. First, a steady-state simulation is performed to find a representative initial temperature distribution. Then, the transient simulation is performed. The HotSpot thermal simulation parameters are listed in Table II.

TABLE III

INTERCONNECT ENERGY COMPUTATION PARAMETERS

| Parameter | Value |
|---|---|
| $E_{bit}^{horizontal}$ [$pJ$] | 0.127 |
| $E_{bit}^{vertical}$ or $E_{bit}^{TSV}$ [$pJ$] | $9.56\times 10^{-3}$ |
| $E_{bit}^{router}$ | 70% of $E_{bit}^{horizontal}$ |

TABLE IV

APPROACHES CONSIDERED FOR COMPARISON

| Approaches | Abbreviation | References |
|---|---|---|
| Load Balanced mapping | LB | [23] |
| Application (App.) Aware mapping | AA | [47] |
| Platform (Plat.) Aware mapping | PA | [35] |
| App. views & Plat. characteristics exploitation | AVCE+PCE | Proposed |
| App. frames & Plat. characteristics exploitation | AFCE+PCE | Proposed |

In our 3-D IC model, the interconnect energy consumption is computed by employing (2), as described in Section III-C. Table III lists the parameters used to compute the interconnect energy consumption. For consistency, all the parameters are considered for the 90-nm process technology node, such that energy consumption can be computed accurately. First, the horizontal link energy per bit, $E_{\text{bit}}^{\text{horizontal}}$, is derived as in [14]. Then, the vertical link energy per bit, $E_{\text{bit}}^{\text{vertical}}$ ($E_{\text{bit}}^{\text{TSV}}$), is calculated by using the parameters from ITRS [22]. For the same process technology node, $E_{\text{bit}}^{\text{router}}$ is ~70% of $E_{\text{bit}}^{\text{horizontal}}$, as shown in [43]. $E_{\text{bit}}^{\text{TSV}}$ is only 7.5% of $E_{\text{bit}}^{\text{horizontal}}$, providing substantial space for communication energy optimization by exploiting the links in the vertical direction.

We present the results obtained from our approach and compare them with relevant existing methodologies, as abbreviated in Table IV. The load balanced (LB) mapping approach try to balance load (power) on the cores to achieve good thermal balance and is employed by setting $c_1 = 1$, $c_2 = 0$, and $c_3 = 0$ in (8). The application aware (AA) mapping and platform aware (PA) mapping approaches exploit application and platform characteristics, respectively. The AA approach looks the communicating actors and tries to map them on the same or neighboring cores, whereas PA approach tries to map hot and cold actors on layers having high and low heat dissipation capabilities, respectively. The resulting mapping obtained by the PA approach is not further optimized, i.e., there is no iterative refinement to achieve a better quality of mapping. Our approaches AVCE + PCE (or PCE + AVCE) and AFCE + PCE (or PCE + AFCE) exploit, respectively, views and frames related applications' characteristics along with the exploitation of architecture characteristics and are carried out by setting appropriate constants ($c_1 = 0$, $c_2 = 1$, and $c_3 = 1$) in (8). Furthermore, our approaches perform iterative refinement to optimize the mapping quality in terms of peak temperature and energy consumption.

### *B. Results for Different Video Sequences*

Fig. 6 shows interconnect power consumption and peak temperature when employing different mapping approaches to map the four considered video (application) sequences on a 4 × 3 × 3 mesh architecture. The interconnect power is

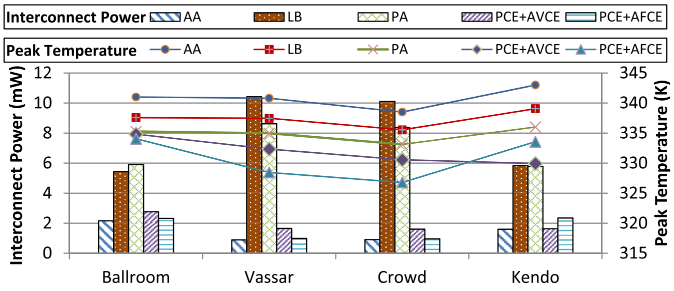


Fig. 6. Interconnect power and peak temperature for different applications.

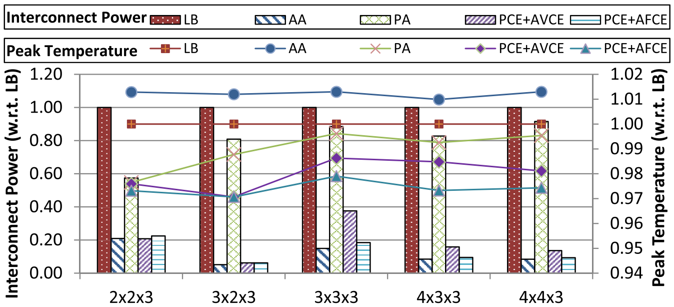


Fig. 7. Interconnect power and peak temperature for different platforms.

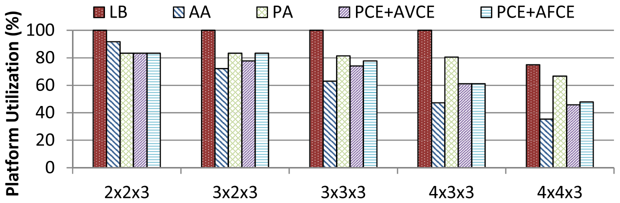


Fig. 8. Utilization with varying platform sizes.

estimated as the average communication energy per second. A couple of observations can be made from Fig. 6.

1) AA approach achieves low interconnect power consumption due to mapping communicating actors close to each other, but results in high peak temperature due to heat stacking in a particular region. The interconnect power consumption by AA and our approaches PCE + AVCE and PCE + AFCE are almost the same.
2) PA approach results in lower peak temperature compared with LB due to platform characteristics exploitation.
3) Our approaches PCE + AVCE and PCE + AFCE results in low energy consumption and peak temperature as they exploit both the application and architecture characteristics and perform iterative refinement to achieve a better quality of mapping leading to lower energy consumption and peak temperature when compared with other approaches. Moreover, the PCE + AVCE shows higher reduction in energy and peak temperature when the application contains higher interview correlations, whereas in the case of higher interframe correlation, PCE + AFCE performs better. On an average, our methodology PCE + AFCE reduces interconnect power consumption by 76% and average peak temperature by 4 °C when compared with PA that provides good results for both power consumption and peak temperature. The reduction in peak temperature is not significant as PA already tries to achieve low peak temperature by exploiting architecture characteristics.

We have measured computation energy as well in order to observe its contribution to the total energy consumption. For the considered video sequences, on an average, the ratio of computation and total energy consumption indicates that the computation energy contributes 74% to the total energy consumption, and thus, the communication energy contribution is 26%. However, since computation energy by all the approaches remains the same due to computations performed in the homogeneous cores, communication energy optimization is the primary focus of our proposed approach. Computation energy can be optimized by employing the DVFS on cores [48], but DVFS is not our focus and orthogonal to our approach. Considering above contributions, our methodology PCE + AFCE reduces total energy consumption by ~6.3% when compared with PA.

### C. Results at Varying Platform Sizes

We analyzed the effect of considering various platforms for the applications on the reduction in energy consumption and peak temperature. Fig. 7 shows interconnect power consumption and peak temperature at various platforms for *Vassar* video when different approaches are employed. The shown values are normalized with respect to LB that leads to worst interconnect power consumption. However, LB provides lower peak temperature than AA. It can be observed that our approaches PCE + AVCE and PCE + AFCE outperform other approaches if both interconnect power and peak temperature need to be optimized for different platforms. Therefore, for any chosen platform that might be required to perform computations, our approach can be employed to achieve good results in terms of interconnect power and peak temperature.

Fig. 8 shows platform utilization in terms of percentage usage of available number of cores for varying platforms. It can be observed that LB shows the utilization of maximum number of cores. The utilization is 100% up to the platform size 4 × 3 × 3. For higher size platforms, LB uses a maximum of 36 cores as the number of actors in the 3-D video application, and thus, utilization decreases. AA shows minimum utilization in most of the cases as it tries to use minimum number of cores by placing communicating actors on the same or neighboring cores. The platform utilization by other approaches that exploit platform characteristics is lower than LB as they try to use cores on layers close to the heat sink (cool layers), and in turn, the cores of hot layers are avoided to be used. However, LB leads to high interconnect power and peak temperature as described earlier.

### D. Energy–Temperature Tradeoff Analysis

The energy–temperature tradeoff points are obtained by exploiting varying amount of correlation between views or frames. The amount of correlation increases by giving high weight to ACE (AVCE or AFCE) in (8). Fig. 9 shows the effect of correlation exploitation available at view (AVCE) and frame (AFCE) levels for mapping *Ballroom* video

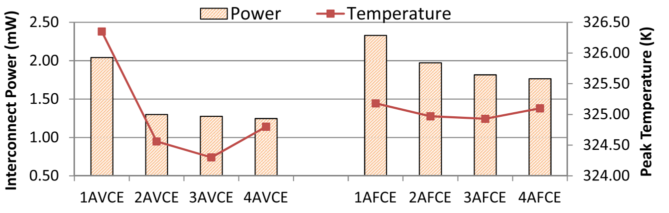


Fig. 9. Effect of correlation exploitation and energy consumption and peak temperature.

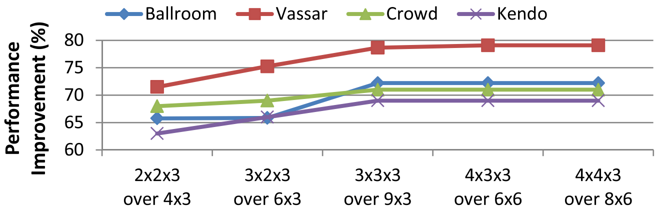


Fig. 10. Performance improvement by 3-D over 2-D architecture.

on 3 × 2 × 3 platform. For higher correlation exploitation, the weight given to ACE in (8) is increased. For example, weight $c_3$ is varied from 1 to 4 when employing AVCE and AFCE, as shown on the horizontal axis. A couple of observations can be made from Fig. 9. First, energy savings increase with the amount of correlation exploitation for both view/frame levels. Second, peak temperature first decreases and then increases in both the cases. The decreasing trend is obtained as a better thermal balanced mapping is being found by exploiting the application characteristics. An increase in temperature takes place as higher correlation exploitation tries to map communication tasks on the same core or stack, which results in nonuniform heat dissipation, and thus, higher temperature.

### *E. Performance Improvement Over 2-D Architectures*

The proposed approach can be applied to various 2-D multicore architectures, and the obtained results can be compared with respect to the 3-D architectures containing the same number of cores. Fig. 10 shows performance (throughput) improvement for different video sequences by 3-D architectures over 2-D architectures when our approach PCE + AVCE is employed. For fair comparisons, the number of cores in both the 2-D and 3-D architectures is kept the same. The improvements are obtained mainly due to the use of vertical links available in 3-D architectures. Vertical links implemented using TSVs have shorter length than the horizontal links, and thus, they provide reduced interconnect distance between vertical adjacent cores. This often leads to faster and more energy efficient communication compared with the horizontal links. Furthermore, in 3-D architecture, we have more neighbors, and hence, fewer hops for overall communication leading to lower latency and in many cases higher throughput. The faster communication leads to low communication time, resulting in reduced overall application execution time. Thus, improvements in the applications throughput are achieved.

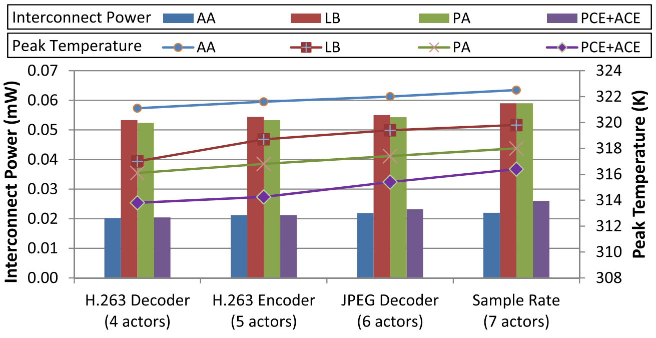


Fig. 11. Interconnect power and peak temperature for different streaming multimedia applications.

It can be observed that the first performance improvement increases with the number of cores due to better usage of cores and then becomes consistent as the number of used cores remains constant. On average, 70% improvement in throughput is obtained compared with a 2-D system-on-chip.

### *F. Generalization: Results for Streaming Multimedia Applications*

Fig. 11 shows interconnect power consumption and peak temperature when employing different mapping approaches to map streaming multimedia applications containing different numbers of actors on a $2 \times 2 \times 3$ mesh architecture. For applying our approach to streaming multimedia applications, the criticality of actors and correlation between them are exploited as application characteristics and platform characteristics are exploited as earlier. A couple of observations can be made from Fig. 11. First, the interconnect power and peak temperature are quite less when compared with 3-D video sequences (Fig. 6) that have complex structure with 36 actors. Second, interconnect power and peak temperature increase with the application size (number of actors) as processing overhead in the network and cores increases. It can be seen that our approach PCE + ACE provides good results to optimize for both interconnect power and peak temperature when compared with other approaches for streaming multimedia applications as well.

For the considered multimedia applications, on an average, the computation energy contributes 72% to the total energy consumption, leaving 28% contribution as the communication energy.

### *G. Offline Overhead*

The offline overhead depends on time to find a mapping, its thermal simulation, and the number of iterations to identify the best mapping in terms of peak temperature and energy consumption. The time to find a mapping is quite small and is in the order of a few milliseconds. The thermal simulation overhead is high and depends upon the spatial grid resolution and the number of layers in the 3-D multicore architecture. Usually, it takes several minutes to find the best mapping due to iterative exploration, where each iteration takes ∼2 min on a 1.70-GHz Intel i5 CPU (single threaded) when a grid resolution of $32 \times 32$ and three layers in the multicore architecture

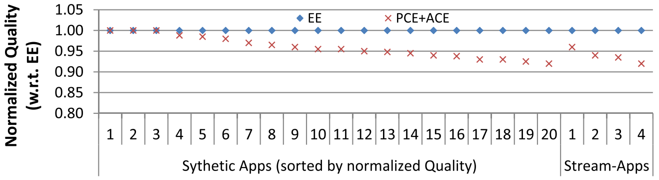


Fig. 12. Quality of mappings by our approach and EE.

are considered. The proposed methodology converges in three to eight iterations for different applications and architectures. This results in a total offline overhead of up to 16 min.

### H. Online Overhead

The online processing for a video sequence starts after the actors of the video application are configured on the platform resources. This configuration is performed once at the application startup for a given video sequence and has a small overhead. For example, PCE + AFCE approach takes a configuration time of 18 ms to configure *Ballroom* video sequence on a 3 × 3 × 3 architecture. For other approaches and architectures, the overheads are of similar orders.

### I. Deviation From Optimal Mapping

To find the optimal mapping, an exhaustive exploration (EE) approach (see [49]) has been employed, which evaluates all the possible mappings (actors to cores allocations) to select the best (optimal) quality mapping for temperature and energy. In order to restrict the evaluation for an application within a few hours, small size synthetic and real-life applications are considered to be mapped on a small size 2 × 2 × 2 architecture. Fig. 12 shows the quality of the mappings for 20 synthetic applications (containing four to seven actors) and four streaming multimedia applications: 1) H.263 decoder; 2) H.263 encoder; 3) JPEG decoder; and 4) sample rate converter by our approach PCE+ACE with respect to the optimal mappings achieved by EE. It has been observed that loss in quality of mappings by our approach is more when the number of actors increases. As can be seen from Fig. 12, our approach provides optimal mapping for some of applications. For the remaining applications, the quality is comparable with EE, and maximum deviation is $<9\%$.

## VI. Conclusion

We present a novel methodology to map 3-D video processing on 3-D multicore platforms. We show that the methodology exploits application and platform characteristics toward achieving energy savings and reduction in peak temperature while satisfying the throughput requirement of the application. The experimental results indicate that our approach can be employed to variety of platforms to achieve high-quality results. In future, we plan to consider heterogeneous cores to be integrated in the 3-D multicore to explore further opportunities for energy savings and peak temperature reduction. In addition, we plan to consider complex 3-D multicore systems containing a huge amount of cores and mapping of multiple applications on them at the same time.

## References

[1] N. A. Dodgson, "Autostereoscopic 3D displays," *Computer*, vol. 38, no. 8, pp. 31–36, Aug. 2005.

[2] K. Müller, A. Smolic, M. Drose, P. Voigt, and T. Wiegand, "3-D reconstruction of a dynamic environment with a fully calibrated background for traffic scenes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 4, pp. 538–549, Apr. 2005.

[3] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[4] FujiFilm Global. *FinePix REAL 3D W3*. [Online]. Available: http://www.fujifilm.com/products/3d/camera/finepix_real3dw3/

[5] *Panasonic HDC-SDT750K*. [Online]. Available: http://www2.panasonic.com

[6] *Joint Draft 8.0 on Multiview Video Coding*, document JVT-AB204, 2008.

[7] X. Xu and Y. He, "Fast disparity motion estimation in MVC based on range prediction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2008, pp. 2000–2003.

[8] Y. Kim, J. Kim, and K. Sohn, "Fast disparity and motion estimation for multi-view video coding," *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 712–719, May 2007.

[9] J.-P. Lin and A. C.-W. Tang, "A fast direction predictor of inter frame prediction for multi-view video coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2009, pp. 2589–2592.

[10] P.-K. Tsung, W.-Y. Chen, L.-F. Ding, S.-Y. Chien, and L.-G. Chen, "Cache-based integer motion/disparity estimation for quad-HD H.264/AVC and HD multiview video coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 2013–2016.

[11] B. Zatt, M. Shafique, S. Bampi, and J. Henkel, "Multi-level pipelined parallel hardware architecture for high throughput motion and disparity estimation in multiview video coding," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2011, pp. 1–6.

[12] R. S. Patti, "Three-dimensional integrated circuits and the future of system-on-chip designs," *Proc. IEEE*, vol. 94, no. 6, pp. 1214–1224, Jun. 2006.

[13] P. Ramm, A. Klumpp, J. Weber, and M. M. Taklo, "3D integration technologies," in *Proc. IEEE Symp. Design, Test, Integr. Packag. MEMS/MOEMS*, Apr. 2009, pp. 71–73.

[14] Y. Cheng, L. Zhang, Y. Han, and X. Li, "Thermal-constrained task allocation for interconnect energy reduction in 3-D homogeneous MPSoCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 2, pp. 239–249, Feb. 2013.

[15] C. Liu, L. Zhang, Y. Han, and X. Li, "Vertical interconnects squeezing in symmetric 3D mesh network-on-chip," in *Proc. IEEE 16th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2011, pp. 357–362.

[16] C. Fabre *et al.*, "PRO3D, programming for future 3D many-core architectures: Project's interim status," in *Formal Methods for Components and Objects*. Berlin, Germany: Springer-Verlag, 2013, pp. 277–293.

[17] B. S. Feero and P. P. Pande, "Networks-on-chip in a three-dimensional environment: A performance evaluation," *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 32–45, Jan. 2009.

[18] K. Bernstein *et al.*, "Interconnects in the third dimension: Design challenges for 3D ICs," in *Proc. Design Autom. Conf. (DAC)*, Jun. 2007, pp. 562–567.

[19] F. Clermidy, F. Darve, D. Dutoit, W. Lafi, and P. Vivet, "3D embedded multi-core: Some perspectives," in *Proc. IEEE Conf. Design, Autom. Test Eur. (DATE)*, Mar. 2011, pp. 1–6.

[20] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a Teraflops processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, Sep./Oct. 2007.

[21] J. Howard *et al.*, "A 48-core IA-32 message-passing processor with DVFS in 45 nm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2010, pp. 108–109.

[22] (2010). *International Technology Roadmap for Semiconductors*. [Online]. Available: http://www.itrs.net/reports.html

[23] X. Zhou, J. Yang, Y. Xu, Y. Zhang, and J. Zhao, "Thermal-aware task scheduling for 3D multicore processors," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 1, pp. 60–71, Jan. 2010.

[24] E. A. Lee and D. G. Messerschmitt, "Static scheduling of synchronous data flow programs for digital signal processing," *IEEE Trans. Comput.*, vol. C-36, no. 1, pp. 24–35, Jan. 1987.

[25] L.-F. Ding, P.-K. Tsung, W.-Y. Chen, S.-Y. Chien, and L.-G. Chen, "Fast motion estimation with inter-view motion vector prediction for stereo and multiview video coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May/Apr. 2008, pp. 1373–1376.

[26] Z.-P. Deng, K.-B. Jia, Y.-L. Chan, C.-H. Fu, and W.-C. Siu, "A fast view-temporal prediction algorithm for stereoscopic video coding," in *Proc. IEEE Conf. Image Signal Process.*, Oct. 2009, pp. 1–5.

[27] L. Shen, Z. Liu, T. Yan, Z. Zhang, and P. An, "View-adaptive motion estimation and disparity estimation for low complexity multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 925–930, Jun. 2010.

[28] N. Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, and T.-S. Chang, "Algorithm and architecture of disparity estimation with mini-census adaptive support weight," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 792–805, Jun. 2010.

[29] M. Shafique, S. Garg, J. Henkel, and D. Marculescu, "The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives," in *Proc. ACM Design Autom. Conf. (DAC)*, 2014, pp. 1–6.

[30] H. Khdr, S. Pagani, M. Shafique, and J. Henkel, "Thermal constrained resource management for mixed ILP-TLP workloads in dark silicon chips," in *Proc. ACM Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.

[31] A. K. Singh, M. Shafique, A. Kumar, and J. Henkel, "Mapping on multi/many-core systems: Survey of current and emerging trends," in *Proc. ACM Design Autom. Conf. (DAC)*, May/Jun. 2013, pp. 1–10.

[32] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," in *Proc. IEEE Conf. Design, Autom. Test Eur. (DATE)*, Apr. 2009, pp. 1410–1415.

[33] C. Addo-Quaye, "Thermal-aware mapping and placement for 3-D NoC designs," in *Proc. IEEE Int. SOC Conf. (SOCC)*, Sep. 2005, pp. 25–28.

[34] C. Sun, L. Shang, and R. P. Dick, "Three-dimensional multiprocessor system-on-chip thermal optimization," in *Proc. IEEE/ACM/IFIP Conf. Hardw./Softw. Codesign Syst. Synth. (ISSS+CODES)*, Sep./Oct. 2007, pp. 117–122.

[35] M. Cox, A. K. Singh, A. Kumar, and H. Corporaal, "Thermal-aware mapping of streaming applications on 3D multi-processor systems," in *Proc. IEEE/ACM/IFIP Workshop Embedded Syst. for Real-Time Multimedia (ESTIMedia)*, Oct. 2013, pp. 11–20.

[36] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph, "Three-dimensional chip-multiprocessor run-time thermal management," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 8, pp. 1479–1492, Aug. 2008.

[37] S. Liu, J. Zhang, Q. Wu, and Q. Qiu, "Thermal-aware job allocation and scheduling for three dimensional chip multiprocessor," in *Proc. IEEE 11th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2010, pp. 390–398.

[38] K. Puttaswamy and G. H. Loh, "Thermal herding: Microarchitecture techniques for controlling hotspots in high-performance 3D-integrated processors," in *Proc. IEEE 13th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2007, pp. 193–204.

[39] K. Kang, J. Kim, S. Yoo, and C.-M. Kyung, "Runtime power management of 3-D multi-core architectures under peak power and temperature constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 6, pp. 905–918, Jun. 2011.

[40] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3D chip multiprocessors using network-in-memory," *ACM SIGARCH Comput. Archit. News*, vol. 34, no. 2, pp. 130–141, 2006.

[41] A. K. Singh, A. Kumar, and T. Srikanthan, "Accelerating throughput-aware runtime mapping for heterogeneous MPSoCs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 18, no. 1, pp. 9:1–9:29, 2013.

[42] University of Virgina. (2011). *Hotspot 5.02 Temperature Modeling Tool*. [Online]. Available: http://lava.cs.virginia.edu/HotSpot

[43] S. Bhat, "Energy models for network-on-chip components," M.S. thesis, Dept. Math. Comput. Sci., Technische Univ. Eindhoven, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2005.

[44] A. H. Ghamarian *et al.*, "Throughput analysis of synchronous data flow graphs," in *Proc. IEEE 6th Int. Conf. Appl. Concurrency Syst. Design (ACSD)*, Jun. 2006, pp. 25–36.

[45] S. Stuijk, M. Geilen, and T. Basten, "SDF$^3$: SDF for free," in *Proc. IEEE 6th Int. Conf. Appl. Concurrency Syst. Design (ACSD)*, Jun. 2006, pp. 276–278.

[46] S. Segars, "ARM7TDMI power consumption," *IEEE Micro*, vol. 17, no. 4, pp. 12–19, Jul./Aug. 1997.

[47] E. L. de Souza Carvalho, N. L. V. Calazans, and F. G. Moraes, "Dynamic task mapping for MPSoCs," *IEEE Des. Test Comput.*, vol. 27, no. 5, pp. 26–35, Sep./Oct. 2010.

[48] V. Chaturvedi, A. K. Singh, W. Zhang, and T. Srikanthan, "Thermal-aware task scheduling for peak temperature minimization under periodic constraint for 3D-MPSoCs," in *Proc. 25th IEEE Int. Symp. Rapid Syst. Prototyping (RSP)*, Oct. 2014, pp. 107–113.

[49] P. Yang *et al.*, "Managing dynamic concurrent tasks in embedded real-time multimedia systems," in *Proc. IEEE/ACM/IFIP Conf. Hardw./Softw. Codesign Syst. Synthesis (ISSS+CODES)*, Oct. 2002, pp. 112–119.

**Amit Kumar Singh** (M'09) received the B.Tech. degree in electronics engineering from the Indian School of Mines, Dhanbad, India, in 2006, and the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2012.

He was with HCL Technologies, Noida, India, for a year and a half. From 2012 to 2014, he was a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. Since 2014, he has been with the Department of Computer Science, University of York, York, U.K. He has authored over 40 papers in leading international journals/conferences in his research areas. His current research interests include system level design-time and run-time optimizations of 2-D and 3-D multicore systems with a focus on performance, energy, temperature, and reliability.

Dr. Singh was a recipient of the PDP 2015 Best Paper Award, the HiPEAC Paper Award, and the GLSVLSI 2014 Best Paper Candidate. He has served as the Session Chair for conferences, such as the Asia-Pacific Embedded Systems Education and Research Conference and the Design, Automation & Test in Europe Conference. He is a Technical Program Committee Member of the IEEE/ACM conferences, such as the International Symposium on Electronic System Design, the International Workshop on Network on Chip Architectures, and the Medicaid Enterprise Systems Conference.

**Muhammad Shafique** (M'11) received the Ph.D. degree in computer science from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, in 2011.

He has over ten years of research and development experience in power-/performance-efficient embedded systems in leading industrial and research organizations. He is currently a Research Group Leader with the Chair for Embedded Systems, KIT. He holds one U.S. patent. His current research interests include design and architectures for embedded systems with a focus on low power, reliability, and adaptivity.

Dr. Shafique was a recipient of the 2015 ACM/SIGDA Outstanding New Faculty Award, six gold medals, the 2011, 2014, and 2015 CODES + ISSS Best Paper Awards, the 2011 Conference on Adaptive Hardware and Systems Best Paper Award, the 2008 Design, Automation & Test in Europe Conference (DATE) Best Paper Award, the 2014 Design Automation Conference Designer Track Poster Award, 2010 the International Conference on Computer-Aided Design (ICCAD) Best Paper Nomination, several HiPEAC Paper Awards, and the Best Master Thesis Award. He is the Technical Program Committee (TPC) Co-Chair of ESTIMedia 2015 and 2016, and has served on the TPC of several IEEE/ACM conferences, such as ICCAD and DATE.

**Akash Kumar** (M'05–SM'13) received the B.Eng. degree in computer engineering from the National University of Singapore (NUS), Singapore, in 2002, the master's degree in technological design with a minor in embedded systems jointly from NUS and the Eindhoven University of Technology (TUe), Eindhoven, The Netherlands, in 2004, and the Ph.D. degree in electrical engineering with a minor in embedded systems jointly from TUe and NUS, in 2009.

He was with the Department of Electrical and Computer Engineering, NUS, from 2009 to 2014. He is currently with the Technische Universitat Dresden, Dresden, Germany, where he directs the Chair for Processor Design. He has authored over 100 papers in leading international electronic design automation journals and conferences in his research areas. His current research interests include design, analysis, and resource management of low-power and fault-tolerant embedded multiprocessor systems.

Dr. Kumar was a recipient of the best paper award nominations, including the Conference on Field Programmable Logic and Applications (FPL) in 2014, GLSVLSI in 2014, SC in 2015, and the Design, Automation & Test in Europe Conference (DATE) in 2015. He is also a Technical Program Committee Member of major conferences in the design automation and field-programmable gate array design area, such as the Design Automation Conference, DATE, the Conference on Automation Science and Engineering, the Asia and South Pacific Design Automation Conference, FPL, and the Conference on Field Programmable Technology.

**Jörg Henkel** (M'95–SM'01–F'15) received the master's and Ph.D. (*summa cum laude*) degrees from the Technical University of Braunschweig, Braunschweig, Germany.

He was with NEC Laboratories, Princeton, NJ, USA. He is currently with the Karlsruhe Institute of Technology, Karlsruhe, Germany, where he directs the Chair for Embedded Systems. He holds ten U.S. patents. His current research interests include design and architectures for embedded systems with a focus on low power and reliability.

Prof. Henkel was a recipient of the 2008 Design, Automation & Test in Europe Conference Best Paper Award, the 2009 IEEE/ACM William J. McCalla the International Conference on Computer-Aided Design (ICCAD) Best Paper Award, and the 2011 and 2014 CODES + ISSS Best Paper Awards. He was the Chairman of the IEEE Computer Society, Germany Section, and the Editor-in-Chief of *ACM Transactions on Embedded Computing Systems*. He is an Initiator and the Spokesperson of the National Priority Program called Dependable Embedded Systems of the German Science Foundation, and was the General Chair of ICCAD in 2013.