Abstract

Network-on-Chips (NoCs) have been widely employed in the design of multiprocessor system-on-chips (MPSoCs) as a scalable communication solution. NoCs enable communications between on-chip Intellectual Property (IP) cores and allow those cores to achieve higher performance by outsourcing their communication tasks. Mapping and Scheduling methodologies are key elements in assigning application tasks, allocating the tasks to the IPs, and organising communication among them to achieve some specified objectives. The goal of this paper is to present a detailed state-of-the-art of research in the field of mapping and scheduling of applications on 3D NoC, classifying the works based on several dimensions and giving some potential research directions.

Index terms— Survey, 3D Network on Chip, 3D NoC, Mapping, Scheduling, 3D System-on-chip, 3D SoC
1 Introduction

Nowadays, technological innovation allows us to seamlessly implement increasingly complex applications on computing systems equipped with powerful processors. Each processor generation outperforms the previous one, which entails incorporating more logic onto the silicon [1]. But there are two issues with this approach. The first is that we are losing the ability to downsize transistors and the logic and memory blocks that are incorporated in them [2]. The second issue is that individual chips have reached their maximum size. Thus new technologies bring the concept of hybrid bonding and integrate an entire system on a single chip, known as System-on-Chip (SoC).

To address the increasing application demands, SoC architecture has become more advanced, evolving from a single processor to a multiprocessor system-on-Chip (MPSoC) [3]. This progress was made possible by the shrinkage of transistors, which allowed for the integration of billions on a single chip [4]. MPSoCs may now accommodate tens of Intellectual Properties (IPs), such as processor cores, memory modules, and other I/O components [5]. These systems boost performance by combining many homogeneous or heterogeneous processors, connected together to form a network.

The Network-on-Chip (NoC) paradigm has evolved as an enabler for the integration of numerous embedded cores on a single device. NoC has been presented as a way to replace traditional interconnections in SoC and MPSoC-based buses, guaranteeing flexible communication and high necessary bandwidth [6]. Because of its regularity and simplicity, the 2D-mesh topology is the most researched and utilised NoC topology. However, when the number of cores rises, resulting in higher network size, network performance might suffer drastically [7]. The adoption of 3D NoC is the proposed method to increase performance in terms of power savings and average hop-count [8][9]. A 3D NoC-based MPSoC is built by stacking numerous levels on top of one other and connecting them vertically via through-silicon vias (TSVs). Since 3D NoC-based MPSoCs can integrate more processors and execute more programs, establishing efficient and accurate mapping and scheduling techniques is necessary. Previous surveys cover various mapping techniques aiming 2D architectures [10][11], while authors in [12] review security threats and countermeasures targeting regular
3D NoC. However, to the best of our knowledge, task mapping and scheduling in 3D NoC are still uncharted areas. Mapping and Scheduling steps follow 3D NoC specialization and their job is to implement the given application into the selected architecture, which primarily means to assign and order the tasks and communications of the application into the resources of the architecture such that the design goals are optimized. Figure 1 demonstrates the scheduling and mapping of an application onto a 3D NoC system. The application is represented as a Directed Acyclic Graph (DAG) where each node indicates some tasks (computation unit). Each task has a weight associated with it that represents its cost of execution. The process is achieved using two steps. At first, the tasks of the DAG are ordered for their execution. This is termed as scheduling. In the figure, the tasks are grouped based on their execution order (numbered as 1, 2 and 3). The next step is the mapping where the tasks are to be allocated to the cores.

Area and power consumption for 3D NoC can be accurately estimated using the findings of latency, throughput, and communication bandwidth obtained from the simulated network. After designing the communication infrastructure, communication technique, and evaluation framework for NoC, the final and most crucial process is to associate and determine the appropriate arrangement and placement of application activities on different cores. This is highly beneficial in reducing 3D NoC’s communication costs and overall performance, and it constitutes the fourth dimension, mapping and scheduling in 3D NoC [13],[14],[15]. We concentrated primarily on this research topic in this study to assess mapping and scheduling methodologies in 3D NoC architectures. The aim of this paper is to provide a comprehensive overview of all the aspects of mapping and scheduling (task graph generation, scheduling, optimization techniques, simulation setup, and performance evaluation metrics). The taxonomy we propose for classifying all the papers given in this work is shown in Figure 2 hierarchically. Our work illustrates mapping methodologies using different categorization than [16], [17] or [18]. In addition, we classify the strategies according to certain optimization metrics. The organization of this survey is as follows: first, we provide the reader with the need and an overview of the mapping and scheduling model in 3D NoC, presented in Section 3. We present a brief overview of the underlying technologies of the 3D NoC in Section 2. In Section 4 the related work of mapping and scheduling techniques in 3D NoC architectures is discussed. Section 5 talks about the optimization of various attributes followed by Section 6 where the experimental details are provided and Section 7 concludes this paper by giving insight into future opportunities.
2 3D Technologies

We need shorter and denser connections to transfer enormous volumes of data on the same chip, which can only be accomplished by stacking one chip on top of another. In a 3D design, connecting two chips face to face means making hundreds or thousands of micrometer-long connections per square millimeter. These small, dense connections allow data to flow from one piece of silicon to another as quickly and efficiently as if the two were on one chip. Microbumps, 3D integration or merging 2.5D and 3D technology via Through Silicon-Vias (TSVs) contribute to certain significant ways for establishing dense connections.

1) 2.5D integration with 3D: In 2.5D structure, there is no stacking of dies on dies, but dies are on Silicon Interposer. The dies are packed into a single package in a single plan and both are flip-chipped on a silicon interposer. Interposers and dies are layered one on top of the other in a 3D framework using TSVs. Intel employs 2.5D technology in combination with 3D by integrating two 3D stacks of chiplets with high-density interconnects [19]. The dies also link high-bandwidth memory and an I/O chiplet to the biggest chiplet (base tile), on which the rest of the chips are placed. The base tile subsequently uses 3D stacking technology to stack compute and cache chiplets atop it. This technology creates a dense matrix of vertical die-to-die connections between two chips.

2) 3D integration: It refers to stacking of IC chips and connecting them using Microbumps or TSVs, and stacking multiple device layers on a single chip, which may or may not use very-fine-pitch TSVs to form the interconnect. Even if one of the chips in the stack lacks a single transistor, 3D integration is believed to speed up calculations. Microbumps are a miniature version of flip-chip packaging technology where solder bumps are added to the endpoints of the interconnects at the
top (face) of a chip. The chip is then flipped onto a matching set of interconnects on a package substrate, and the solder is melted to make a connection. To stack two chips using this approach, the first chip must have small copper pillars protruding from the surface. These are then coated with a solder microbump, and the two chips are linked face-to-face by melting the solder. TSVs are copper interconnects that run vertically down the silicon of a chip. They do not span the whole depth of a wafer, hence the backside of the silicon has to be ground away until the TSV is exposed. In 3D stacked chips, the chips are bonded together such that their interconnects are face-to-face. So, the TSVs supply the stack with power and data access. Graphcore implements 3D integration by connecting a power-delivery chip to its AI processor \[20\]. The power-management die is densely packed with capacitors and TSVs.

3) Hybrid bonding: Copper bonding: Hybrid Bonding connects copper pads at the top of a chip’s interconnect stack to copper pads on another chip. The pads in hybrid bonding are in small recesses surrounded by an oxide insulator. When the copper pads of two chips are placed opposite to each other at normal temperature, the insulator is chemically activated and bonds instantly. The copper pads expand in an annealing stage and bridge the gap to generate a low-impedance link. It offers significant bandwidth and power benefits over state-of-the-art microbump-based approaches. AMD’s Zen 3 with 3D V-Cache \[21\] is a 3D stacked device that uses hybrid bonding to attach additional cache to a high-performance processor.

3 3D NoC Mapping and Scheduling Model

3.1 3D NoC Architecture

3D-MPSoC is represented as a regular mesh of tiles built across many layers and linked via Network-on-Chip (NoC). It can be seen in Figure 3. Every tile has a processing core that is connected to a network router. Routers are connected together through links, which constitute the overall architecture. Packet-mode communication is employed in the architecture, hence all the data transferred in the network is loaded into packets, and all packets are split down into small pieces known as flits. Each core block is further subdivided into compute blocks and L2 cache \[22\]. Lateral communication is carried out via NoC, while vertical links are implemented using TSVs.

3-D NoC architecture is impacted by three factors: 1) Topologies, 2) Router designs and 3) 3-D NoC Design methodology

1) Topology determines how the nodes in the network are connected with each other \[23\]. In a multiple-hop topology, packets may travel via one or more intermediary nodes before reaching the destination node. Mesh and torus topologies are common multiple-hop topologies used in 3D NoCs. The authors in \[24\] present a three-layer NoC with a unique topology for each layer to fulfill its individual cost-performance requirements, and routers are connected by crossbar switches in the same pillar but at separate layers. In terms of network performance and energy dissipation, Feero et al. \[9\] compared and examined 3-D mesh-based topologies (symmetric 3-D mesh, stacked mesh, and ciliated mesh) and 3-D tree-based architectures.

2) Router Designs: Scalable NoC contains a router at each hop to control access to shared resources \[25\]. The following is how a conventional NoC router operates. The router first writes a received packet into a buffer and computes the packet’s admissible output port at the same time. The Virtual-Channel (VC) allocation mechanism then assigns the packet a free virtual channel of the output channel. When a VC is obtained successfully, the packet arbitrates for accessing the crossbar ports. If everything goes well, the packet can proceed to a neighbor router via the switch.
traversal (ST) and link traversal (LT) stages. Going through these stages sequentially generates significant delay on each hop, making communication inefficient. Furthermore, if any of the desired resources (e.g., buffer, crossbar) are not available, a packet is stalled, increasing the packet delay. To enhance the performance and reduce the manufacturing cost of 3D NoCs with minimal distortion to the modularity, inhomogeneous architectures have been proposed to merge 2D and 3D routers in 3D NoCs [26]. However, inhomogeneous 3D NoCs, have a performance trade-off due to the restricted number of 3D routers. So, authors in [27] proposed a co-design of routing algorithms and router architectures to mitigate the limitations of NoCs in heterogeneous 3D SoC. Another efficient three-stage pipelined adaptive VC router is proposed by combining the low-complexity bypassing technique with adaptive routing to balance the traffic in hybrid NoCs to achieve low-latency communication under various traffic loads [28].

3) Design Methodologies: Various designs for the efficient 3D NoC architecture have been presented by modifying or merging topologies, routers, or vertical links. In 2D NoC, data packets are transferred between distant cores, where the number of hops increases the delay substantially. To address the issue of interconnect delay, 3-D integration is adopted. TSV is a well-known 3D integration method but they occupy a large area on the chip and also lead to cost increase due to additional fabrication processes and low yield. To tackle these issues, wireless interconnect using capacity coupling and inductive coupling is an excellent substitute for TSVs. Joseph et al. [29] use inductive coupling for application-specific 3D NoC design that reduces the design complexity caused by channel interference while increasing channel efficiency. Next, in order to improve the overall system power, performance, and reliability of these mesh architectures, we also need to address routing issues and buffer utilization. Rahmani et al. in [30] introduce an architecture that has a higher tolerance for single-bus failure compared to existing architectures. Dynamic power decreases quadratically as supply voltage decreases, hence using several supply voltages is a well-known low-power solution. Siozios et al. [31] show that a single power supply for designing homogeneous NoCs is inefficient and introduces a 3D NoC architecture with multiple supply voltages. In addition, a high-level mapping algorithm for facilitating application mapping onto such NoCs is presented where the applications are clustered into groups and these groups are mapped
onto the proposed architecture.

Performance and energy consumption are prime concerns in today’s Systems-On-Chip (SoCs) due to the continuous scaling of process technology. Scaling permits design engineers to integrate as much IP as possible on a chip in order to conduct multitasking and meet client requirements. However, it leads to reduced interconnect performance and scaling issues have become a constraint for multiprocessor system-on-chip communication. As a result, authors in [32] propose an effective design and scaling of on-chip interconnect architectures which is divided into four steps. The initial stage is to identify the IP cores and their size, as well as the core’s probable mapping with regard to the router. The core size is equal to the total number of IP cores present in the MPSoCs after clustering, which will be mapped onto the topology. The number of mapping possibilities is equal to the number of routers in the topology architecture, which determines the number of generations. The second step is to estimate the fitness function, which yields the total energy consumption (EC) and communication costs (CC) for mapping the IP cores onto routers. The third stage determines the local best core and global best core. Every core has a local best, which is the permutation of core locations with the lowest EC and CC values among all permutations of that core. For a certain transmission, the IP cores with the lowest network energy and CC values are the global best of that transmission. The final stage is the evolution over many generations where a swap operator and a swap sequence are introduced through which the local best and global best are updated, which contribute to the discovery of the optimal solution for the lowest network energy and CC based mapping. Extension of NoC architecture to three dimensions, brings benefits of both approaches together, meaning more performance of communication, better scalability, and lower power consumption. The last one is due to the shorter wire length and interconnects capacitance. However, despite all these benefits, a critical dilemma is intensified at higher integration levels. As device density increases, power density increases too and as a result, thermal management and required cooling solutions become more challenging. To address the aforementioned constraints, we explore several mapping and scheduling techniques for regular 3D NoC-based MPSoCs.

3.2 Mapping and Scheduling Model

Mapping and scheduling may be performed both online and offline. Offline or static mapping and scheduling are conducted prior to application run-time. Since static mapping and scheduling are executed once during compilation, they have no effect on application performance at run-time. Online or dynamic mapping and scheduling entail the assignment and sorting of tasks while the application is running. This should result in a better solution, but the computational complexity of mapping and scheduling techniques may increase the application’s latency and energy consumption at run-time.

The primary focus of our research here is to provide details about scheduling and mapping techniques in 3D NoC systems. While doing so, we also need to talk about routing to some extent because routing significantly impacts the effectiveness of scheduling and mapping. Since communications between cores in a regular NoC are achieved by routing messages hop-by-hop from the source to the destination, it is essential to include routing algorithms in the discussion of mapping and scheduling for the effectiveness of 3D NoC architectures. Thus, along with considering papers on scheduling and mapping, we also covered some routing strategies. This is shown in Figure 4.
3.2.1 Mapping Model in 3D NoC

The 3D NoC mapping platform is composed of resource nodes, routing, and physical links. Network mapping on a chip is the process of assigning logical IP cores with known communication relations and traffic volumes on a given Task Characteristic Graph (TCG) to the corresponding resource nodes of the Architecture Characteristic Graph (ARCG) under specified system constraints, allowing various applications to be completed smoothly and efficiently under the constraints [33].

3D NoC mapping algorithms can be divided into two categories: i) the conventional mapping optimization techniques employed in the early days and ii) new intelligent mapping algorithms based on heuristics. Traditional mapping methods primarily comprise the exhaustive technique, linear programming algorithm, and branching algorithm, and they are designed to tackle problems when the early structure of the on-chip network is basic and their application is rare. The majority of current research is focused on intelligent algorithms, which are search mechanisms designed to solve optimization problems by mimicking natural laws or biological characteristics, such as the genetic algorithm, ant colony algorithm, bat algorithm, tabu search algorithm, particle swarm algorithm, and simulated annealing algorithm. Traditional mapping methods produce generally accurate results, but as application scenarios get more complicated, the amount of calculation has become excessive. Meanwhile, it is even difficult to find the final result based on the existing computing capabilities. Therefore, there is no need to find all the solutions, as long as the intelligent algorithm that provides better and more practical solutions is immediately recognized by the majority of researchers. Many diverse mapping strategies are provided in a range of publications that use both types of algorithms separately or in combination to get a viable answer. We explore various mapping techniques such as Application mapping, Task mapping, and Core mapping (Figure 4) and study them through an algorithmic perspective.
3.2.2 Scheduling Model in 3D NoC

The time spent on computation and communication to execute an application is a key component in assessing its performance. While the calculation time is primarily dictated by the architecture of the IP cores, the communication time is determined not just by the routing protocol but also by how the tasks are scheduled. The execution order of tasks must be established to achieve optimized performance. This is referred to as the scheduling problem which can be done using a scheduling algorithm. Scheduling is the act of creating a schedule, which is a timetable for planned occurrences. The process of mapping answers the question 'where', but to answer 'when', scheduling is required. Scheduling is the time ordering of tasks and communications on their assigned resources, which assures the mutual exclusion between executions of tasks on the same resource. If multiple numbers of tasks of an application are mapped onto one core, then task scheduling is encountered. Given an application task graph mapped onto 3D NoC architecture, scheduling is the time ordering of tasks and communications determining the order in which tasks and transactions between them are to be executed such that deadlines (for real-time tasks) are met and some pre-defined parameters are optimized. A scheduling problem can be thought of as either a constraint satisfaction or a constrained optimization problem. We can define a scheduling problem by:
1. A set of time intervals defining activities, processes, or tasks to be done.
2. A set of temporal limitations, i.e. definitions of potential connections between the start and end time of the intervals.
3. A set of specialized constraints i.e. description of the complicated interactions on a set of intervals due to the state and finite capacity of resources.

A suitable scheduling approach is one that satisfies design requirements such as timing constraints, data precedence constraints, and memory size constraints. It should also optimize the design goals such as reducing energy usage, improving timing performance, and balancing memory utilization. Many scholars have employed distinct strategies for the same, such as using deterministic and heuristic methods, while others have incorporated mathematical models such as Integer Linear Programming (ILP), Non-Linear Programming (NLP), and Mixed Integer Linear Programming (MILP). We group these algorithms in four types of scheduling categories as shown in Figure 4. (1) General Scheduling (2) Task Scheduling (3) Test Scheduling (4) Allocation-based Scheduling.

4 Mapping and Scheduling: Algorithmic Survey

Previous works have mainly focused on mapping and scheduling techniques in design/run time or optimization parameters. However, to the best of our knowledge, none covers the mapping and scheduling methods from an algorithmic standpoint. Authors in [34] presented a nice survey of the scheduling techniques concerning real-time tasks. Our work is structured around several algorithms as shown in Figure 5. We have discussed various techniques that cover exact/deterministic methods that always give optimal solutions and approximate methods that are further divided into approximation and heuristics. All the papers that adopt the aforementioned algorithms are included in Table 1.

4.1 Mapping

One of the most crucial parts of designing an NoC structure is mapping. It is the process of determining the arrangement of IP cores on the NoC structure. Here, we have classified existing works
Figure 5: Classification based on Algorithms
Table 1: Algorithms used in papers covered

| Algorithm        | Technique                          | Reference Papers |
|------------------|------------------------------------|------------------|
| Constraint       | Programming                        |                  |
| Backtracking     | ILP                                | [35], [36], [37], [38] |
| Bottom-To-Top    |                                    | [43], [44]       |
| Branch and X     | Branch and Bound                   | [45], [46], [47] |
| Queuing          | Highest level first                |                  |
|                 | Highest Priority First             |                  |
|                 | Earliest Deadline First            |                  |
| Greedy           | Balanced min k-way partitioning    | [51]             |
|                 | Dijkstra’s algorithm               | [52]             |
| Local Search     | STAGE                              | [53], [54]       |
|                 | Approximation                      | [55]             |
| Evolutionary     | Genetic Algorithm                  | [56], [52], [57], [55], [58], [59], [60], [61], [62], [63], [64], [65], [66] |
|                 | Hybridised Constructive Heuristic  |                  |
|                 | (HyCH)                             |                  |
| Swarm Intelligence| Bat Algorithm                      |                  |
|                 | Particle Swarm                     |                  |
| Trajectory based | Simulated Annealing                |                  |
| Meta-Heuristic   |                                    | [62], [65], [74], [75] |
| Fuzzy Algorithms | Fuzzy Logic                        |                  |
| Voltage Based    | Algorithm                          |                  |
|                 | Voltage and Frequency Scaling      | [77], [78]       |
| Memetic Algorithm| Linear Search                      |                  |
| ML Based         | Clustering                         |                  |
| Lottery Algorithm| Lottery Arbiter                    |                  |

based on algorithmic paradigms as shown in Figure 5.

Heuristics under approximate methods (Figure 5) provide a good estimate of the solution, but we do not know how accurate it will be (intuitive). They are used to solve NP problems and decrease the time complexity of problems by giving quick solutions. An evolutionary algorithm is one such heuristic-based approach used for solving problems that cannot be easily solved in polynomial time, such as classically NP-Hard problems, and anything else that would take far too long to exhaustively process. One type of evolutionary algorithm is the Genetic algorithm (GA), which is used for mapping and scheduling in a wide range of research papers. GA is considered as an efficient algorithm for the mapping in NoC structures. A genetic algorithm creates an initial, random pool of viable solutions (chromosomes), which are assessed on each iteration (generation) by assigning fitness scores based on an objective function that steers the drive toward an efficient solution to the issue [83]. Authors in [56] employ GA for the placement of application tasks on 3D NoC structure targeting three optimization types - thermal aware, communication aware, and hybrid optimization. The hybrid optimization approach, targeting an optimal trade-off between communication and thermal aware methods, reduces communication significantly which in turn reduces the power consumed by the chip. As a result, in addition to the reduction in communi-
cation, the average temperature of the chip in this approach is very close to that obtained in the initial thermal-aware method. Moreover, since the peak temperature is also reduced, this approach allows the mapping of multiple tasks on each processing element, which gives a significant boost in performance. A power-aware mapping for allocation of IPs over mesh nodes is proposed in [52] where GA is used to determine the network mapping that uses the least amount of power, while the Dijkstra algorithm is used to identify the shortest routing path. The authors introduce $P_{\text{link}}_{ij}$ and $P_{\text{router}}_{ij}$ to represent the power consumed by links and routers in 3D-mesh networks. Next, a power matrix $P = [P_{ij}]$, where $P_{ij} = P_{\text{link}}_{ij} + P_{\text{router}}_{ij}$ and a weight matrix $W = [w_{ij}]$ that uses Dijkstra algorithm is created to find the least cost path in a graph enabling exchange of data between all IPs. The weight matrix $W$ also helps in deriving the accurate power model of 3D NoC. Furthermore, the objective function, total power consumption $P_{\text{total}}$ is minimized.

If more than one metric is examined in any case, a multi-objective genetic algorithm provides a better result. SOGA (single-objective genetic algorithm) is a heuristic search technique that is based on natural selection and evolution [84]. Instead of (SOGA), a rank-based multi-objective genetic algorithm is applied in [57] to minimize energy usage and packet latency of 3D NoC. In this method, three operators i.e selection, crossover, and mutation are defined which use real number encoding instead of traditional encoding where binary data is used as chromosomes. The selection operator is based on ranks of chromosomes instead of fitness and authors access the packet latency under two conditions: no congestion and congestion. The algorithm identifies the nondominant fronts and eliminates their nodes. Thus, it finds an approximate pareto front. However, this scheme completely neglects the temperature of 3D NoC. Another GA-based mapping approach has been presented in [35] which optimizes the energy by mapping the task graph nodes onto the stacks, then the stacks are mapped onto the bus, and finally, the bus’s position is modified to reduce transmission energy. Likewise, authors in [59], [63] also employ a GA to reach the chip’s thermal equilibrium and for the reduction in both temperature deviation and peak temperature.

A further approach to mapping problem is to formulate it using an integer linear programming (ILP)-based technique [85] which is a type of exact method (Figure 5). Hamedani et al. [39] use ILP and propose three mapping strategies to analyze thermal limitations and their influence on the temperature and performance of a 3D NoC. The first algorithm minimizes the peak temperature of the chip under the constraints of bandwidth, performance, and task assignment. However, the complexity of this mapping increases exponentially with the number of variables which is in the order of $|PE| \times |E|$, where PE is processing elements and E is the edges of the communication task graph. Therefore this model may not be efficient for the next generation of chips. The second algorithm proposes a mapping of tasks based on hierarchical partitioning using ILP with the aim of minimizing cuts between partitions. It takes into consideration, the constraint of bandwidth, task assignment, and power consumption. Although the first algorithm takes more information into account, the second algorithm produces final results quite rapidly. Therefore the third algorithm called approximation mapping was proposed which uses the second algorithm for partitioning and the first for merging. It shows a significant reduction in peak temperature.

Effective run-time mapping on such 3-D NoC-based MPSoCs can be difficult since the arrival order and task graphs of the target applications are often unknown in advance, which is exacerbated further by strict energy constraints for NoC systems. Hence, authors in [86] propose a run-time incremental mapping that reduces energy consumption by classifying the application as communication-centric or computation-centric. If it is communication-centric, edges with heavy data flow are mapped vertically. If it is computation-centric, high-power-consumption jobs are routed closer to the heat sink.
A summary of general mapping techniques is illustrated in Table 2.

Mapping is further divided into application mapping, core mapping, and task mapping. Application mapping is a process of assigning tasks to the IP cores, whereas core mapping is the process of allocating IP cores to optical routers for a particular communication purpose. In task mapping, the tasks are organized in accordance with a criterion before being mapped into IP cores. We have summarized the algorithms used in different types of mapping as shown in Table 3.

4.1.1 Application Mapping

Application mapping has evolved into a critical component of 3D NoC/3D SoC architecture design. It performs by assigning application-related tasks to specific cores in the first instance, and then an efficient mapping technique is adopted to calculate the optimal arrangement of these cores on the nodes of NoC. The application mapping approaches must be implemented keeping in mind the constraints of many-core on-chip systems, such as bandwidth, communication time, latency, throughputs, power, and energy. Authors in [43] introduce a three-step mapping algorithm where initially a 3D cuboid core region of a specific shape is selected for each application. Secondly, the exact locations of the core regions in the chip are determined, followed by a task-to-core mapping. Finally, during the task running, the defragmentation process is executed. Wadhani et al. [45] present a Branch-and-Bound heuristic for smart application to core mapping in 3D NoC architecture.

We know that an application is implemented on a set of collaborative (IPs). The selection of the best-suited block from a library of IPs, as well as their physical mapping into the 3D NoC architecture, are both NP-complete problems. In this case, multi-objective solutions are preferred. Authors in [91] introduce Differential Evolution to deal with the blocks mapping problem in order to implement efficiently a given application which is then broadened to multi-objective optimization in order to decrease the final implementation’s hardware space, execution time, and power consumption. Bougherara et al. [70] also uses Multi-Objective Particle Swarm Optimization (MOPSO) to produce the optimal IP and physical mapping of a particular application on three-dimensional topologies. The majority of research papers are concerned with the mesh-based topology of 3D NoC. CastNet3D [42] is a network for application mapping onto mesh-based 3D-NoCs. The study proposes an ILP formulation as well as an innovative heuristic method with the goal of minimizing energy consumption. However, in [79] in addition to mesh topology, other topologies such as torus, ring, and Butterfly Fat Tree (BFT) are used for effective mapping. They used three models to analyze the effective topology: power, area, and delay. A Knowledge-based Memetic Algorithm (KBMA) is also developed to achieve an appropriate trade-off between the cost function and the design of the 3D NoC.

4.1.2 Task Mapping

The assignment of tasks to the available IP cores is called Task Mapping. In 3D NoC, each node is connected to a router through a network interface. Hence, the task mapping stage can simply be augmented to decide on the particular core to be used. As connections between cores in a traditional NoC are accomplished by routing messages hop-by-hop from the source to the destination, a multi-hop traversal is enabled by leveraging efficient router-bypass methods. Router data paths, in particular, may be set statically as well as dynamically to enable multi-hop traversal, allowing flits to completely avoid intermediary router pipelines, resulting in ultra-low latency performance. The performance gains can only be completely achieved if there is no contention among flows that share
| Ref. | Algorithm | Target for optimization | Strategy applied | Objective | Limitation |
|------|-----------|-------------------------|-----------------|-----------|------------|
| 43   | Search tree and Defragmentation | TSV integration | Application to Core Mapping | Optimizing communication latency and temperature | i) Shows worst performance when system is extremely fragmented or when applications have no communication among the tasks. ii) No support for non-contiguous mapping |
| 56   | Genetic Algorithm | TFG and PE | Mapping of logical tasks onto a single PE | i) Reduces power consumption ii) Reduces volume of messages transferred | i) Increases the temperature of system ii) Storage overhead due to encoding |
| 59   | ILP (Integer Linear Programming) | Interconnects and Topology | Mapping of tasks onto different PEs | Optimizing peak temperature of the chip | The complexity of the mapping problem increases exponentially with the number of variables Only the peak temperature of the chip is considered. The distribution of temperature fragmentation is neglected. |
| 52   | Genetic Algorithm | Topology and IP cores | Mesh network mapping | Optimizing power consumption of the chip | i) Very high Computational complexity ii) Communication latency and temperature of the chip are neglected |
| 57   | Rank-Based Multi Objective Genetic Algorithm | Routers, IP Cores | Mesh network mapping | Optimizing the packet latency | i) The strong elitism may incur high selection pressure, which leads to premature convergence. ii) They support mapping only for single application systems, ignoring the capacity of 3D-MPSOC to handle several applications during the execution time. |
| 35   | Backtracking and Genetic algorithm | TSV and Flits | Mapping IP cores on buses | Optimising the Communication energy | i) Execution time increases due to filtering of bad results ii) Other constraints like latency and temperature of mapping application onto the chip are neglected. |
| 74   | Simulated Allocation (SAL) | EM-induced TSV, Routers | Core to tile mapping | Optimising the reliability of TSVs in the 3D chip | Does not satisfy the energy along with reliability requirements |
| 65   | Cataclysm Genetic-based Simulated Annealing (CGSA) | IP cores, Topology | Application to core mapping | Finding the best IP core mapping solution along with the best matched 3D NoC topology | i) Fabrication Cost overhead ii) Premature stagnation |
| Ref | Algorithm | Target for optimization | Strategy applied | Objective | Limitation |
|-----|-----------|-------------------------|-----------------|-----------|-----------|
| 45  | Branch and Bound | Cores and Routing paths | Application to core Mapping | Reducing dynamic communication energy consumption | Only limited to core mapping in regular NoC |
| 79  | Memetic Algorithm and Linear Search | Topology | Mapping of IP blocks to 3D NoC tiles | Analysing the efficient topology for mapping based on multiple objectives (total power consumption of the chip, total area of the chip, total delay in the chip) | No consideration of Network conditions |
| 87  | ARTEMIS | Mapping Design | Runtime framework for efficient dynamic application mapping | Enhance the performance and lifetime of 3D NoC-based chip multiprocessor | Does not consider the temperature of system |
| 70  | Multi Objective Particle Swarm Optimization (MOPSO) | IP Blocks and Topology | Mapping to application on different topologies | Selection of the most suited IP block | Does not support fault tolerance |
| 88  | Multi-Objective Immune algorithm | IP cores | Application to core Mapping | Determining the best mappings and optimising latency and power | No consideration of other performance models like area, delay and reliability |
| 76  | Fuzzy logic | IP cores | Task to core mapping | Optimizing the temperature, communication delay and power consumption of the chip | Does not fully exploit the vertical links to optimize latency |
| 89  | Partitioning and Categorization | TSV | High volume task to low volume task mapping | Optimization of network latency | Does not consider thermal aspects. |
| 58  | Genetic algorithm | IP cores | Task to core mapping | Optimize communication energy and peak temperature | Does not consider thermal aspects. |
| 60  | Exhaustive exploration, genetic algorithms | IP cores | Task to core mapping | Optimising the peak temperature and performance | Does not consider fragmentaion which leads to high communication latency after running many applications. |
| 77  | DVFS | 3D Multi cores | Task to core Mapping | Optimising temperature of the chip | Network latency are not considered |
| 66  | Hybridised Constructive Heuristic (HyCH) | TSV | CTG mapped on the PCTG (Physical Communication Topology Graph) | Improve Performance i.e. less communication overhead, execution time | Focus only on the regular topologies |
| 51  | Balanced mm | TSV | Task to core mapping | Reducing communication energy | No consideration of temperature of system |
| 90  | Dual | 3D routers | Application to core Mapping and IP cores to tile mapping | Optimising the average packet latency and communication energy | Increase in average hop-count and delay |
| 69  | Particle Swarm Optimisation | TSV | Mapping of cores to routers | Decreasing the communication cost | Higher runtime for realistic use cases. |

Table 3: Classification of algorithms based on Mapping Types
common connections along their routing pathways. When there is conflict, bypass pathways must be terminated early, and the corresponding flits must be halted and buffered at intermediate routers for arbitration, degrading to hop-by-hop communication in the worst-case scenario. So, authors in [92] address this contention problem by introducing SMT (Satisfiability Modulo Theories)-based contention-free task mapping and scheduling framework where an application task graph can be statically compiled to a multi-core or parallel processing platform for non-preemptive execution based on 2D and 3D. Manufacturers are now lowering the amount of on-chip TSVs or employing 3D mesh-based NoCs with partially-filled TSVs to boost yield [93]. However, 3D NoCs with partially filled TSVs have irregular topology. Ziaeeziabari et al. [89] propose a mapping algorithm for irregular topologies and introduce the terminology such as application graph, topology graph, and ranking parameter to answer the problem of task mapping in 3D NoC. The application graph deals with the set of tasks and the communication between them whereas the topology graph deals with the set of cores and channels connecting the cores. The ranking parameter for each task in the application graph can be expressed by the equation,

$$\text{Ranking}(t_i) = \sum_{\forall j=1,2,3,..|T|,i\neq j} v(d_{i,j}) + v(d_{j,i}),$$

where $v(d_{i,j})$, $v(d_{j,i})$ are volumes of communication tasks. The algorithm has four phases 1) Partitioning of application graph into high and low volume communication 2) Mapping multi-task partitions of high-volume communications 3) Mapping multi-task partitions of low-volume communication 4) Mapping single-task partitions.

Another set of works considers the clustering of tasks for mapping. Clustering is a method of grouping a collection of tasks so that tasks in the same group are more related to those in other groups. A cluster-based thermal-aware task allocation technique is presented by [80] which maps the tasks of an application into clusters to optimize the temperature. However, as the number of tasks increases, so does the complexity of the algorithm. So, genetic algorithms are introduced for efficient task mapping. Shen et al. [58] propose a GA for thermal-aware task mapping on 3D NoCs. The suggested task mapping technique is known as 3D-TTM, and it is divided into two stages: communication-aware group mapping and thermal-aware task scheduling. Zhu et al. [60] define a similar technique with two stages: communication-aware group (CAG) and thermal-aware scheduling (TAS). TAS is implemented at run-time, whereas CAG is implemented at design-time. The optimal mapping is selected and super tasks are developed using GA or exhaustive exploration during the CAG stage, meanwhile, the super tasks created in the CAG stage will be mapped to the cores in different layers during the TAS step.

Dynamic voltage and frequency scaling has recently been used to address heat issues on 3-D NoC architectures. By decreasing core operating voltages, DVFS can efficiently manage the temperature of 3-D MCPs. Voltage scaling [77],[78] has also been used for mapping in 3D NoC architectures. Thermal-aware mapping and voltage scaling (TAMVES) is proposed where task-to-core mapping is followed by voltage scaling [78], whereas authors in [77] propose a two-stage approach that combines design-time mapping and DVFS with run-time thermal optimization. Another important decision to be taken for task mapping is, among multiple cores which core is the best candidate that satisfies all the requirements. In such a situation fuzzy logic may be used that acts like human decision-making. Hence, Mosayyebzadeh et al. [76] employs fuzzy logic and maps the tasks on the basis of the linguistic rules having linguistic variables such as heat transfer, distance from the source core, and distance from the hotspot.
4.1.3 Core Mapping

Core mapping is one of the most challenging problems in the world of 3D NoC which is the placement of cores within the optimal accessible space of the 3D chip in the context of application-specific 3D Network-on-Chip systems. Authors in [36] intend to tackle this problem with a single-stage constraint programming (CP) technique. Given a Communication Task Graph (CTG) and subsequent task allocations for the cores, heterogeneous CPU cores are assigned to the best available locations on the chip to reduce total communication costs across cores. Concurrently, the application scheduling stage is executed to select the best core types from a set of technological possibilities and to reduce the makespan, or time required to perform all compute jobs on CTG. Guo et al. [65] solve IP-core mapping and topology selection problems using Cataclysm Genetic-based Simulated Annealing (CGSA) considering 3D Optical Network-on-Chip (ONoC). In CGSA, the GA is integrated with an enhanced simulated annealing algorithm and cataclysm strategies to accelerate the searching process. In addition, to improve network reliability, CGSA is coupled with topology selection, i.e., CGSA generates the ideal mapping solution based on the best-matched 3D ONoC topology.

3D NoCs with heterogeneous architecture integrates 3D and 2D routers. However, due to the restricted amount of 3D routers and vertical links, mapping applications in 3D NoCs is difficult. So, Agyeman et al. [90] propose a technique of mapping applications to tiles such that the total communication energy is minimized for which the dimensions of NoC are determined. It was observed that symmetric NoCs have low energy consumption, average packet latency, and average hop count. Clusters are formed by grouping tasks with close communication dependencies such that each cluster contains at least one task with high communication bandwidth so as to balance traffic and reduce hotspots. Two lists are used – ClusterMasters containing tasks with high communication volume and ClusterSlaves for mapping the tasks at vertices and their neighboring tiles respectively. As we know, TSVs occupy a significant area and do not shrink at the same rate as gates, therefore, Manna et al. [69] propose a TSV placement to reduce the TSV count. They use Particle Swarm Optimisation (PSO) to map routers on a partially vertically-connected 3D mesh NoC with cores of different sizes. The assignment of the routers to the vertical links is determined by particle structure and fitness function which is based on the shortest communication distance.

4.2 Scheduling

Scheduling is a mechanism that helps the device to arrange a certain factors like tasks or traffic in a particular order for the efficient performance of the device. Our device is 3D-NoC. round-robin (RR), first-Come First-Served (FCFS), GA, bottom-to-top, highest level first with estimated time (HLFET), and Exhaustive exploration are some of the most widely used scheduling algorithms in 3D NoCs.

A summary of scheduling techniques used in 3D NoC and its types is illustrated in Table 4.

The requirement for scheduling techniques is driven by the need for MPSoCs to support real-time applications. Network topology, router architecture, and routing algorithms are three major operations to consider while designing a 3D NoC. Applications demand proper predictability of execution and communication times, and the router must efficiently use the limited bandwidth of the links to transmit communication data between processing elements while meeting the various criteria for each service level. Hence, a decentralized scheduler design is proposed by [49] that resolves conflicts between traffic flows sharing one or more router output ports. It supports Round Robin and Highest Priority First scheduling algorithms that are utilized for best-effort and guar-
| Scheduling Type | Ref. | Algorithm | Target for optimization | Strategy applied | Objective | Limitation |
|-----------------|------|-----------|-------------------------|------------------|-----------|------------|
| General         | 49   | Round-Robin and Highest Priority First Scheduling Algorithms | Router packets | Decentralized scheduler | Resolving conflicts between traffic flows, sharing one or more router output ports | Difficult to find a correct time quantum during run-time scheduling |
| Task            | 44   | Bottom-To-Top | Run-time scheduling of task graphs | Task to core scheduling | i) Optimizing the peak temperature ii) Reducing execution time of applications | Communication distance is ignored, which might lead to a higher network latency |
|                 | 48   | Highest Level First with Estimated Time (HLFET) | Processors of 3D-NoC | Scheduling of application within cluster of processors | Eliminating the thermal hotspots in the chip | Network latency of the chip is not considered |
|                 | 60   | Exhaustive exploration, Genetic algorithms | IP cores | Tasks to core Mapping in a scheduling interval | Optimizing Communication latency and peak temperature of 3D system | Numerous thermal simulations for task allocation significantly reduce efficiency |
|                 | 77   | DVFS | 3D Multi cores | Task to core mapping and scheduling | Optimizing the peak temperature of the system | i) Swapping overhead due to swapping of tasks ii) Only considers the power balance of processors ignoring the thermal balance |
|                 | 78   | Voltage scaling | 3D Multi cores | Task to core mapping and scheduling | Optimizing throughput and energy consumption while satisfying the thermal constraints | Computation-intensive which leads to large online overhead ii) Slow and Execution time increases by Layer-By-Layer Mapping |
|                 | 50   | Earliest deadline first | Multicore chips | Scheduling of tasks onto layers | Reducing the heat of the chip | Qualitatively allocated hot tasks to cores that are near to the heat sink rather than quantitatively estimate the thermal loads of processors |
|                 | 94   | Rotation scheduling | IP multi-cores | Task to core scheduling | Optimizing the peak temperature of the chip | Concentration only on reducing the temperature of the top layer |
|                 | 95   | Approximation and probability | 3D Multi cores | Task to core mapping and scheduling | To optimize temperature at a low performance cost | i) No consideration of leakage/temperature dependency in the analysis. ii) No consideration of data dependencies in an application |
| Test            | 96   | Design-for-testability (DFT) | Packets and routers | Test Packet Delivery to IP core | Optimising the chip temperature during testing | The scan architecture cannot provide pattern-independent low-power testing. |
|                 | 97   | Unicast-Based Multicast | IP cores | Test Packet Delivery to IP cores | Optimizing the peak temperature of chip ii) Reducing test time | Disregards Network failures when delivering test packets |
anteed service traffic flows to resolve conflicts between router input ports and affect the network performance. Furthermore, we have divided the existing works into the following categories based on the factors they considered.

### 4.2.1 Task Scheduling

When numerous tasks are given to a single processor, the process of selecting the appropriate or high-priority work is referred to as task scheduling. Various scheduling strategies are mentioned in this division, which simplifies certain challenges encountered by an NoC and also aids in performance enhancement.

Many applications with high-performance requirements are modeled as task graphs. The high computation complexity of these task graphs limits their use in run-time situations. Different works are concerned with the placement and sequencing of tasks during application execution \cite{44,77,50}. They introduce techniques that are used to handle dynamic workload variation in applications at run-time. Zhang et al. \cite{44} propose a Bottom to Top scheme which is designed for run-time scheduling of task graphs. It is an efficient task graph scheduling algorithm that initially maps the task graph to the bottom layer of the 3D-NoC and then it adjusts selectively a few tasks to the top layer to reduce execution time while also maintaining low operating temperature. The thermal management process is complex as there are heat effects of vertically stacked cores and there is difficulty in balancing performance requirements and overheating. Tsai et al. \cite{50} introduce a dynamic thermal-management framework under a modern OS that consists of two parts: Task partitioning and Scheduling. Task partitioning consists of a power-aware partition algorithm where the tasks are sorted in descending order of the peak power and then the tasks with higher peak power are assigned to the bottom cores. Following power-aware partitioning, the thermal predictor is used to compute the temperature of each core in the set of vertically aligned cores, which leads to estimating the peak temperature of the stack. Next, the Earliest Deadline First (EDF) rule is used for task scheduling, with cores arranged in decreasing order of power consumption, thermal resistance, and cooling quota, and task speeds sorted in decreasing order. The results inferred that overheating is prevented without the Quality of Service guarantee being violated. Chaturvedi et al. \cite{77} combine design time mapping and DVFS (Dynamic Voltage Frequency Scaling) with run-time thermal optimization for the optimization of the peak temperature of the system.

Task scheduling is considered an effective approach for eliminating the thermal hotspot without introducing any hardware overhead. However, centralized thermal-aware task scheduling algorithms for 3D-NoC have been limited for incurring high computational complexity as the system scales. So to overcome this, a distributed agent-based thermal-aware task scheduling algorithm for 3D-NoC is proposed in \cite{48}. The algorithm organizes the 3D-NoC processors into clusters, and application scheduling is done simultaneously within each cluster. Three types of software agents handle all schedule procedures. The authors also develop a centralized scheduling algorithm using the same heuristics as their distributed approach for comparison. The distributed algorithm achieves lower peak temperature without increasing the execution delay and the scalability of the distributed algorithm also significantly outperforms the centralized algorithm. Some researchers offer OS-level thermal-aware techniques based on task scheduling. An effective temperature management method with OS-level thread scheduling is introduced in \cite{98} based on 3D Mesh multi-core system. A data-aware task scheduling for each thread on a cool processor is proposed. The main aim of the algorithm is to select the optimal thread to schedule. It has an advantage over conventional and random scheduling. Optimization of system performance is done using this algorithm.
under temperature constraints. Another OS-level thermal prediction model for a 3D chip is proposed in [94] where a task scheduling algorithm based on rotation scheduling aims to reduce the peak temperature on the chip. Rotation Scheduling to Minimize Temperature (RSMT) addresses data dependencies in applications and can generate schedules that effectively reduce peak temperatures of cores by up to 8°C and thus improving the thermal management process.

4.2.2 Test Scheduling

With the increasing demand for embedded cores, the implementation of an efficient architecture of core communication is becoming the new bottleneck of SoC performance. A system-on-chip designer embeds cores that are not manufactured, and hence are untested [99]. To increase the testing efficiency of IP core in 3D NoC, research on 3D NoC test scheduling under many constraints is being done. A possible issue with 3D NoC-based test access is the formation of hotspots owing to stacking and the high toggling rates associated with structural test patterns utilized for industrial testing. Hotspots and high temperatures can cause the system components to fail as it is reported by Feng et al. that with an increase in every 10°C, the failure rate of any hardware component increases by double [100]. This hampers the overall system performance. To eliminate hotspots, Xiang et al. [96] present a thermal-driven test scheduling strategy in which the entire NOC bandwidth is utilized to transmit test packets. A Unicast-based multicast scheme [97] has also been introduced which avoids delivering test packets to hotspots in the NoC by utilizing the homogeneity of the cores.

4.2.3 Allocation-based scheduling

3D NoCs are prone to exhibit severe thermal problems due to high power density and long heat dissipation paths in the vertical directions [101]. Hotspots can arise quickly in 3D NoCs because of the uneven temperature distribution, resulting in a shorter chip lifespan and worse system dependability. We have divided this category based on the allocation of jobs or tasks. So we need to balance the temperature among different cores present in 3D NoC architectures. One method for balancing the temperature distribution is to assign and migrate tasks between the cores. It not only enhances system performance while keeping the temperature limit the same, but it also consumes less hardware implementation cost. Researchers have used the terms task allocation or job allocation for various scheduling algorithms.

Adapt3D [95] investigates DVFS and workload migration/scheduling techniques for 3D systems. This technique evenly distributes application loads taking the location of the cores into account and balances the rise in temperature by assigning tasks according to the thermal probabilities of cores. A value for each core is calculated using both the thermal index of each core and thermal behaviors. The temperature history of each core is considered and a thermal index is assigned to each core to differentiate the position of the cores. If the thermal index is low, the core is less prone to hotspots, indicating that it is closer to the heatsink. The thermal probability of each core is determined, and new workloads are allocated to cores based on their probability values. These values are regularly updated at scheduled intervals using a weight factor and the thermal index. Adapt3D performs nearly as well as the default system setup (dynamic load balancing: the default task scheduling policy in modern operating systems), but significantly lowers hotspots. Furthermore, when Adapt3D is paired with DVFS, hotspots are decreased by an additional 20-40%.

Next Liu et al. [102] optimize the temperature of the chip by allocating hotter jobs to the cores which have less thermal resistance whereas the colder jobs are allocated to cores that have high
thermal resistance in order to take the heat off from the chip. This is done by sorting the jobs in the descending order of their power dissipation, then calculating the estimated temperature of the core after executing a specific job, and then the job is allocated to the core only if the estimated temperature is less than the critical temperature otherwise it will be allocated to the next core (cores will be sorted in the ascending order of their thermal resistances). If the temperature of the chip is not balanced then the jobs with low power dissipation will be allotted to the cores with small thermal resistances. Another thermal aware scheduling approach where cores are allocated according to temperature is introduced in [55]. The authors use an approximation approach and propose an online-task allocation algorithm, where the tasks are sorted in the descending order of their powers and then an incremental update of thermal simulation is performed so that the same task is allotted to cores (approximation approach). Then the task is allocated to the core which has the lowest temperature rise. Similarly, all the tasks are allocated.

A thermal-aware optimization is proposed in [103] for 3D MPSoCs where the first part of the proposed approach is a power balancing algorithm in which the tasks are prioritized in the order of Earliest Finish Time (EFT) and then the energy consumed by the tasks assigned to a processor is calculated and then the task is assigned to the processor which has a less thermal impact. The second part of the proposed approach is slack distribution and voltage scaling. Task slack distribution is used to permit voltage scaling, which facilitates temporal power density and peak temperature minimization. In the first step of slack distribution, the slacks of all tasks are calculated and then the slacks are grouped into slack paths, then paths are arranged in the descending order of slack, and then a slack is assigned to tasks based upon the values of task execution time, thermal weighted energy required for a task, total path energy consumption and then in the last step all task slacks are computed again and then the voltage of the task is scaled. The third part of the proposed approach is an Iterative hotspot mitigation algorithm in which the thermal hotspots are eliminated. This is explained in Algorithm [1].

Algorithm 1 Iterative hotspot Mitigation

1: while peak temperature of 3D MPSoC can be reduced do
2:     Compute peak temperature and find critical task set within the region of peak temperature
3:     Adjust the slack distribution between the current task and its neighboring tasks.
4:     Calculate the possible peak temperature reduction for each specified task
5:     for each task do
6:         Increase slack time of the task
7:         Decrease slack time of task’s parent and child tasks by half
8:     end for
9:     Recompute peak temperature and record local slack adjustment
10:    Record new peak temperature and then restore initial slack distribution
11:    Employ the slack adjustment with the lowest peak temperature
12: end while

Another scheduling approach is to combine clustering with allocation. Shen et al. [80] introduce a cluster-based thermal-aware task allocation algorithm (CTTA) where every single task is considered as a cluster. The CTTA algorithm is divided into three stages: clustering stage, cluster allocation stage, and in-stack adjustment. The clustering stage is further divided into two steps: initial clustering and cluster combination. Firstly, computation energy, total computation energy, and average computation energy of initial clustered tasks are calculated and in the next step, the
cluster with minimum computation energy is merged with the cluster which has high communication energy with it. The merging takes place if the sum of the computation energy of the two clusters is less than the average computation energy else another cluster that has a second less computation energy is selected and merged with the cluster that has less computation energy. In the initial clustering, a list of tasks that are ready to be executed according to the descending order of their task levels is recorded and the task with minimum task level is fetched and it is clustered with its best predecessor as a new cluster. If the best predecessor is not found then the task itself becomes an individual cluster. In this way, remaining all tasks are clustered. In cluster allocation, the cluster with the largest communication traffic is allocated to the center core and the next cluster which has the maximum communication traffic with the allocated cluster is allocated to the next core. Similarly, all other clusters are allocated to the clustered core nodes. In in-stack adjustment, the task with high computational energy is moved to the core with minimum computation energy to reduce the temperature of the system.

4.3 Routing

A 3D-IC chip has many physical planes, and all IP cores may be integrated on a single physical plane or over multiple physical planes. If an NoC is utilized in a 3D-IC, it may be implemented as either a 2D-NoC (i.e., routers are implemented on a single physical layer) or a 3D-NoC (i.e., routers are implemented in multi physical plans). This is shown in Figure 6. In 3D-NoC, a router on one physical plane could be connected to routers on the same plane or neighboring physical planes [52]. One of the significant design tasks to consider while creating a 3D-NoC is the implementation of an efficient routing algorithm. Different routing algorithms and techniques that combine mapping with routing are present in 3D NoC. Look ahead XYZ routing algorithm [37] is the pipeline design of the traditional XYZ routing algorithm. The virtual channels are not considered in this method for the enhancement of high traffic. This algorithm improves throughput by utilizing four pipeline systems: buffer writing (BW), routing calculation (RC), switch arbitration (SA), and the crossbar traversal stage (CT). As each hop travels through the four phases, the performance suffers and the flit latency rises. The information from the previous stage can be collected in the pipelining system, and the operations from each stage are dependent on the preceding stage. This dependence exists in the RC and SA phases, and it is eliminated in the Look ahead XYZ by having the RC and SA stages parallel.
Many routing algorithms are considered unsuitable for NoC applications which have hotspots or network as priority requirements \cite{104,105}. To address this issue, earlier algorithms included extraneous buffers into the heavy communication channels to distinguish the precedence of each channel, however extra buffers necessitate more resource space and hence consume more power. Appathurai et al. \cite{81} suggest using a lottery arbiter in place of round robin arbiter as it can set the priority for different requests and does not require an additional buffer in the case of heavy communication. Another challenge with routing algorithms is the variable latency and throughput in each layer of a 3D interconnects. Joseph et al. \cite{27} proposed a co-design of routing algorithms and router architectures (two routing algorithms called $Z^+ (XY)Z^-$ and $ZXYZ$) to mitigate network performance due to varying throughput and latency of NoCs. Horizontal and vertical communication is modeled independently because communication within a layer is synchronous whereas communication between levels is not necessarily synchronous. The proposed models provide relevant information on their potentials and set parameters of the routing algorithms. Two routing algorithms are presented based on two principles that consider the movement of packets over multiple layers. These algorithms, when combined with router architectures, claim to have low area overhead and outperform the state-of-the-art in both theoretical and practical evaluations.

The performance of a 3D NoC and the amount of power it consumes, both are impacted by the topology of the network. BFT topology has been implemented in various routing algorithms to boost the performance of NoC architectures. A low latency energy efficient BFT-based 3D NoC design with zone-based routing strategy has been introduced in \cite{106}. It alleviates the upward traffic load problem that occurs in classical BFT especially, with a significantly large network size. Similarly, A 3D topological design of an NoC based on the BFT topology is described, together with a newly proposed table-based routing mechanism in \cite{107}. Another power-aware mapping is proposed by authors in \cite{52} that combines two algorithms. Elmiligi et al. \cite{52} propose to use Dijkstra’s algorithm to find the shortest path routing and GA is used to find network mapping.

## 5 Optimization of Attributes

In this section, we will review the literature on various optimizations of 3D NoC. Here we consider five major categories for optimization parameters that are kept as objective while performing scheduling and mapping of applications on 3D NoC - i) Communication which consists of communication energy and communication latency, ii) Latency which consists of the packet and network latency, iii) Energy, iv) Throughput and v) Temperature.

### 5.1 Communication

Data-intensive applications impose a heavy load on NoC, and communication consumes more time than computation by integrating multiple cores on the chip. Due to the increase in the volume of data and control traffic among the cores, the communication energy, and communication latency problem is addressed through various mapping techniques for communication-aware 3-D NoC designs which use different (Genetic, Heuristics, Branch or Bound) algorithms for finding optimal mapping sequence. If the whole overall communication architecture is optimized according to the traffic patterns generated by the applications under consideration, the achievable performance of a 3D NOC can be enhanced.
### Table 5: Optimization attributes along with some nominal research papers

| Attributes                        | Reference Papers |
|----------------------------------|------------------|
| Communication energy             | 45,90,31,68,58,80,51 |
| Communication latency             | 76,108,37,70 |
| Network latency                   | 89 |
| Packet latency/delay              | 90,41 |
| Latency between the links of NoC  | 57 |
| Power                             | 76,98,37,30,82,52,81,67 |
| Energy                            | 35,78,86,47,109 |
| Performance / Throughput          | 98,60,110,30,87,107,51 |
| Overall temperature               | 76,98,102,36,74 |
| Peak temperature                  | 44,176,58,80,60,77,94,103,42 |
| Average Chip Temperature          | 39,102 |

#### 5.1.1 Communication Latency

A crucial feature of an efficient network design is the ability to guide packets across the network in order to boost communication throughput and minimize latency. Authors in [43], [108] propose a runtime communication-aware mapping algorithm to optimize performance under the thermal constraint in 3D NoCs. Wang et al. [108] introduces an algorithm to optimize latency and overall application running time by finding 3D cuboid core region and determining the location of core regions, followed by task-to-core mapping. It further extends the work in [43] and introduces a defragmentation algorithm to reduce system fragmentation. Bing et al. [43] focuses on reducing communication latency through defragmentation in three steps: (1) For each and every application, the shape of the core region is found using search techniques and it is characterized on the basis of MD (minimal distance to the heat sink) and NOL (number of occupied layers) values. (2) Location of the core region is determined and task-to-core mapping is done. To find the exact locations of the core regions, an algorithm is proposed which scans X and Y directions based on the given corner and starting point to find the location. Mapping tasks to cores is done on the basis of MD, NOL values of tasks, and availability of free cores. (3) Performing defragmentation which includes determining the migration destination for the application and then finding the path to that destination. To reduce fragmentation thermal-aware defragmentation concept was proposed which focuses on migrating applications to chip corners, keeping the free cores clustered in the center. The defragmentation algorithm is triggered only when the fragmentation metric goes above the threshold fragmentation.

#### 5.1.2 Communication Energy

The energy consumed within the system is mainly because of the communication between PEs and the computation performed by PEs. Accordingly, mapping and scheduling techniques play a vital role in determining the amount of energy consumed due to communication. As the industry moves towards many-core chips, architecture designers face challenges related to communication issues. So researchers try to optimize the communication energy using various clustering methods. A novel 3D NoC architecture is proposed in [31] with low-energy features consisting of two layers. Agyeman et al. [90] optimize the communication energy by determining the NOC dimensions for
the application and then clustering the tasks followed by mapping the clusters onto the tiles of the 3D NoC. Similarly, authors in [80] also cluster the tasks and map them onto the core stack which is denoted as a “clustered core” and then move the tasks on the clustered cores to reduce the temperature and reduce communication energy. Other algorithms use Balanced min K-way partitioning and Heuristic algorithms for the same [51]. To optimize the communication energy in the chip, authors in [51] propose an architecture for 3D NoC that consists of two layers where the top layer is powered by a high supply voltage ($V_{dd}$), while the bottom layer is driven by a low supply voltage ($V_{Ldd}$) and then applications are clustered into groups and these groups are mapped onto the proposed architecture.

Grouping the tasks and executing them in batches can be another way to optimize communication energy. To reduce communication energy, authors in [51] use ”Balanced min K-way partitioning and Heuristic Algorithm” where the tasks in the same X, Y coordinates in a 2D plane are grouped into a partition and then communication volume between these groups is minimized. Bhardwaj et al. [68] aims to obtain Pareto optimal IP mappings by optimizing the communication energy using Octahedral traversal Attractive - Repulsive PSO. In [45], it is proposed to use a search tree to identify an optimal mapping with the least amount of communication cost, and the applications are ranked based on their communication requirements. A preference queue containing legal nodes available for branching is maintained in the order of the lowest cost. Until an optimal solution is obtained, the next unexpanded node is selected from the preference queue and the unmapped application with the highest communication requirement is mapped. Each generated child node from this expansion is inspected and pruned if its communication cost or lower limit cost is higher than the upper limit cost. The length of the preference queue, however, affects the runtime.

5.2 Latency

Optimizing the latency is of paramount importance for a computing system. Different well-established mapping and scheduling algorithms have been introduced in order to reduce the communication, network and packet latency [43], [89], [90], [57], [108] in the context of 3D NoC.

5.2.1 Packet Latency

Packet latency is the time delay between when a packet of data goes from one place to another. To decrease average packet latency and energy consumption, tasks with high communication requirements should be mapped as closely to each other as possible. Agyeman et al. [90] considers heterogeneous architecture to improve the performance and latency of 3D NoCs by combining 3D and 2D routers. Each layer contains tiles which are composed of processing cores and routers connected by a network interface. It introduces an efficient core mapping technique that aims to map the applications with minimum communication energy. Initially, it determines the size of NoC (3D NoCs with variable dimension sizes under uniform traffic patterns) and then performs architecture matching. Following that, for the mapping process, application tasks are assigned to NoC regions in the order of their communication volumes, so that high-energy consumer tasks are assigned to as many 2D routers as possible while retaining a specific number of 3D routers to improve performance. Compared to existing algorithms like Onyx [111], CastNet [112] and Nmap [113], the proposed algorithm generates 3D NoCs with low packet latency and energy consumption. However, the average packet latency of Branch-and-bound and the proposed algorithm is almost
the same, So further research can be done on decreasing the average packet latency compared to all other algorithms.

As network congestion may have a large impact on packet delay, two different models are employed for packet latency under no congestion and congestion conditions, which use rank-based multi-objective genetic algorithm (RMGA) [57]. It encodes the different mapping alternatives in chromosomes and obtains optimal pareto-front of the mapping problem. The mapping problem is formulated as a function \((G \rightarrow T)\) where \(G\) is the application characterization graph (APCG) with each vertex referring to the IP core and \(T\) is the architecture characterization graph (ARCG) with each vertex referring to a node of NoC. The map function maps the IP core to the node of NoC. The two models will determine the average packet latency of two conditions (under no congestion, under congestion) respectively. Elitism, crossover, and mutation operators are used to explore the mapping space (developing new mapping alternatives). The mapping alternatives are evaluated through an analytical model by using real number encoding for chromosomes and the selection operator depends upon the rank of chromosomes which is calculated using the following equation.

\[
\text{Rank}(x, t) = 1 + nq(x, t)
\]  

where \(nq(x, t)\) refers to the number of solutions dominating solution \(x\) at generation \(t\). In this algorithm non-dominant fronts will be identified and nodes on those fronts will be eliminated, and then the elastic selection is adopted to select the nodes. Thus, RMGA efficiently succeeds in finding an approximate pareto-front and optimizing the packet latency.

5.2.2 Network Latency

A mapping algorithm attempts to find a core on the chip for each task in order to reduce overall hop count, power consumption, bandwidth needs and network latency. The irregularity of 3D NoCs has a significant impact on 3D mapping techniques. This means that the mapping technique cannot take into account the same communication capabilities for all chip cores. In fact, in a 3D NoC, certain topologically near cores may have greater distance than other topologically distant cores. So, an algorithm that reduces network latency and minimizes hop count, 3D-AMAP is proposed for 3D NoCs with partially filled TSVs having irregular topologies in [89]. The algorithm follows four phases of task mapping - (1) Application graph partitioning, where average communication volume of the application graph is compared with the communication volume between tasks, and communications are categorized as High and Low volume communications. (2) High Volume Multi-task Partitioning, where the task with a high ranking is first mapped on a layer having sufficient free cores followed by mapping of tasks having high communication volume with an already mapped task until all the tasks are mapped. (3) Low Volume Multi-task Partitioning, where task partitions follow a similar process as (2) after being sorted according to their total intra-partition communications in descending order. (4) Single Task Mapping, which maps tasks in the descending order of communications to an empty core having minimum hop count. The algorithm ignores low-volume communications and reduces traffic in TSVs which in turn reduces network latency.

5.3 Power or Energy

With the increasing number of IP cores integrated into the processor, the power consumption of the chip continues to escalate. In 3D NoC, power consumption has a significant impact on system performance, hence, reducing power consumption has become a crucial design consideration for
3D NoC. To meet the ever-increasing communication and low-power demands, 3D NoC designs must be optimized for power and energy consumption.

5.3.1 Power

Chip designers lack the tools required to efficiently execute their applications at various levels of the design hierarchy. To obtain the greatest performance, a design technique for low-power 3D-NoCs applications is required. Elmiligi et al. [52] use GA to determine the optimal 3D-NoC mesh network mapping that consumes the least amount of power for a certain application. They claim to have solved the optimization problem in less than four minutes, compared to an exhaustive search that took three days and still did not identify the lowest power consumption. Dijkstra’s Algorithm is used to find the shortest path routing and GA is used to find the optimum mapping with minimum power consumption for a given application. A bat algorithm (BA) is introduced in [67] for the energy-aware mapping technique for 3D NoC. It requires an improved control strategy to switch between exploration and exploitation in the accurate instant and it needs proper parameter tuning for a better search. In an NoC, the routers consume the majority of energy, and energy consumption increases nonlinearly with the number of input ports. Karthikeyan et al. [82] uses the Lottery algorithm to reduce the power consumed by asynchronous 3D NoC by distinguishing different priorities of the input port and ensuring that it responds to the higher priority port.

5.3.2 Energy

The popularity of battery-driven devices brings the limitation of power availability into consideration, making energy-aware mapping an important research area. Although improvement of energy reduction techniques will facilitate continuous performance betterment, it is seldom the sole optimization criterion. This leads to a trade-off between energy consumption and other user requirements like performance and quality of the solution. It is known that vertical links for communication nodes in 3D NoC are shorter than horizontal ones, therefore they are quicker and use less energy. Hence, Nalci et al. [42] minimize the energy consumption of application by utilizing vertical links for communicating nodes as much as possible. Wang et al. [86] uses dynamic energy technique to reduce the energy consumption which is generated due to the data communication between the processing cores. It classifies the application as communication centric or computation-centric after it specifies a certain condition. If the application is communication-centric then the communication edges which have high traffic volume are mapped in vertical dimensions so that the dynamic energy consumption is reduced. The authors compare the proposed algorithm to branch and bound (BNB) [114] and two heuristic algorithms (Temperature balance (TB) [115] and Temperature-Aware Low Power Mapping (TL) [116]). The experiments show that their proposed algorithm reduces the computational and communication energy more efficiently. Another algorithm that reduces the dynamic communication energy consumption is listed in [63]. It is an energy-efficient mapping algorithm that employs the BNB strategy. A Traffic Equilibrium Mapping method is proposed in [35] for the energy minimization in the chip of 3D NoC-Bus mesh architecture.

Liao et al. [78] aim to optimize the energy consumption by Task-to-core mapping which is followed by voltage scaling. It specifies three components for selecting tasks to be scaled for decreasing temperature and energy: group-power computation (GPC), heat-dissipation rate (HDR), and power-driven temperature reduction (PDTR) and satisfies the thermal constraints by calculating a
where $GPC_i$ denotes the group power of the cores containing the running task, $HDR_i$ denotes the heat-dissipation rate of the core with running task and $PDTR_i$ denotes the power reduction after scaling task in the schedule.

5.4 Throughput

Throughput refers to the performance of tasks by a computing service or device over a specific period of time. Throughput in 3D NoC refers to the number of packets that successfully pass from source to destination paths [117]. The topology of a network affects the performance and power consumption of 3D NOC. So, to get an unprecedented performance gain and optimize throughput by combining the benefits of NoC and 3D ICs, Bose et al. [107] introduce a new 3D topological NoC design which is based on the BFT topology with an efficient table based uniform routing algorithm for 3D NoC. Throughput can be increased only if the temperature of the system is under control. Authors in [78], [55] optimize the throughput by satisfying the thermal constraints. Liao et al. [55] presents an approximated task assignment technique to enhance 3D multi-core processor by rewriting temperature equations to provide an incremental thermal update. The incremental update strategy is used to first determine the current temperature of each core. Then, among the unassigned cores, a new arrival job is assigned to the core with the lowest temperature rise. The above process terminates when all new arrival tasks are assigned to cores. Authors in [78] optimize the throughput by allocating tasks with larger power to upper-layer cores which are closer to the heat sink. As cores at the layer farther from the heat sink dissipate heat more slowly, authors prevent overheating by not assigning tasks to these cores.

Techniques via Task Scheduling [98],[60],[118] have been proposed for optimizing the performance of 3-D NOCs which also satisfy thermal constraints. Fu et al. [98] design a simulator and select the coolest thread using a temperature controller. Then, a data-oriented task scheduling for each thread on a cool processor is done and the algorithm aims to select the optimal thread to schedule. Whereas Zhu et al. [60] present a two-stage scheduling algorithm, namely the communication-aware group (CAG) stage and thermal-aware scheduling (TAS) stage which is implemented at design-time and at run-time respectively. [118] also introduces a run-time thermal management technique (RTM) which uses the greater thermal coupling among the 3D NoC’s vertically stacked routers. A proactive strategy is utilized to optimize performance with varying temperatures where a traffic-aware downward routing shifts horizontal routing power to the bottom layer and adjusts traffic volume to avoid packet congestion.

[87] present a runtime aging-aware application-mapping framework called ARTEMIS for 3D NoC-Based Chip Multiprocessor. It enhances the performance and lifetime of 3D NoC-based chip multiprocessors and manages aging in circuits and the power delivery network (PDN) simultaneously.

5.5 Temperature

Although 3D ICs have many advantages like decreasing interconnect power, scaling of chip dimensions etc., it also deals with various thermal problems. 3D ICs also have high power density which increases chip temperature that further leads to performance degradation. In this case mapping
can be used to control chip temperature. Hamedani et al. [39] reduce the temperature by partitioning the task graphs into subgraphs and mapping the subgraphs using ILP. Wang et al. [98] use task scheduling for optimizing the temperature and performance loss of the chip. Another thermal-aware task scheduling policy consisting of two stages is proposed in [60] which exploits the application and system architecture characteristics to decouple the mapping of task graphs for the performance and peak temperature optimization into two stages namely communication-aware group stage and temperature-aware scheduling stage.

5.5.1 Peak Temperature

The heat generated in the chip is dissipated through a heat sink. The dissipation of heat generated in layers farther from the heat sink is much more difficult. Therefore novel Bottom-to-Top (B2T) task scheduling scheme is proposed in [44] which aims to reduce the peak temperature of the chip by keeping power consumption in such layers lower than those closer to the heat sink. When the part of power consumed by farther layers reduces, less heat is generated in them which results in a lower peak temperature of 3D NoC.

The performance of 3D NoC can be boosted if we reduce the peak temperature as well as optimize the network competition. Since a router can be connected to more than one IP core, this leads to a network competition problem where multiple IP cores and neighbor routers may compete for the same output port of a router. A temperature and network competition-aware mapping algorithm [64] is proposed to reduce the peak temperature and decrease the network competition. This algorithm realizes the multi-objective mapping and ensures a lower time complexity. In the first step of the proposed algorithm, IP cores are clustered according to the number of layers on the chip and the network nodes connected to the same router are clustered together. The second step is to implement a GA for NoC mapping. Simulation results show that this method achieves an appropriate balance between peak temperature and network competition. Authors in [64, 80] use clustering algorithms for the peak temperature optimization.

5.5.2 Average Chip Temperature

The increase in the number of the transistors and higher operating frequency of the processors increase the chip power consumption, which results in a higher power density in the chip, which in turn raises the chip temperature. Hamedani et al. [39] proposes three mapping algorithms and employs ILP to reduce the chip temperature where it partitions the task graphs into subgraphs and then maps the tasks after merging the subgraphs according to their communication rate. Liu et al. [102] also propose three mapping algorithms and optimize the temperature of the chip by allocating hotter jobs to less thermal resistance cores and colder jobs to high thermal resistance cores. This is accomplished by sorting the jobs in descending order of their power dissipation, then calculating the estimated temperature of the core after executing a specific job and allocating the job to the core only if the estimated temperature is lower than the predefined threshold, otherwise, the job is allocated to the next core.

6 Experimental/Evaluation Tools

Let us now discuss the experimental setups researchers typically consider while evaluating the work on scheduling and mapping. The choice of simulators and tools is of great importance to
assess and validate the results. Great attention is paid to record the performance metrics and implementing new architectures and designs for the mapping in 3D NoC architectures. This is the reason why various simulators and benchmarks in the techniques reviewed have been used to evaluate throughput, power consumption, or latency and for measuring the performance of mapping and scheduling in 3D NoCs.

### 6.1 Simulators

Figure 7 shows different simulators used for implementing or evaluating the performances and analyzing the effectiveness of scheduling and mapping techniques in 3D SoC/NoC architectures. Cycle Accurate Simulators such as Noxim\[119\], BookSim\[120\], WormSim\[121\] is used in 33% of the considered papers, full system simulators namely Nirgam\[122\], GEM-5\[123\], GARNET in 15%, and Access Noxim\[124\] constitute 10% of the total. Thermal simulator i.e HotSpot tool \[125\] is used in 25% of the works for temperature modeling. It is usually combined with another tool for a complete evaluation.

### 6.2 Benchmarks

The experiments on the surveyed methodologies make use of a diverse collection of benchmark applications as shown in Figure 8. A considerable portion of the works we analyzed employ both synthetic and real benchmarks. For evaluation of mapping algorithms 31% of total considered papers use real Multimedia benchmarks such as JPEG-encoder, MPEG 4, VOPD, MWD VOPD, DVOPD, PIP, and GSM Coder/Decoder. *Embedded System Synthesis Benchmark Suite (E3S)* as automotive, consumer, networking, telecommunications and office automation constitute 18% of mapping techniques. SPEC benchmarks such as *SPEC CPU 2000* and *SPECjbb* are used in 31% of articles to evaluate most of the scheduling algorithms. 3D stacked SoCs use benchmark circuits like d695, d281, f2126, and 2f2126 for the test scheduling.
PARSEC [126] and SPLASH-2 [127] benchmarks have been widely used in [43], [108], [87], [128], [129], [130], [54]. PARSEC benchmarks namely, dedup, vips, fluidanimate, canneal, and BodyTrack (BT) and SPLASH-2 benchmarks, namely, fft, adix, lu, and water, were used in [43], [130] to explore and establish performance-energy-reliability trade-offs for 3D small-world NoC (SWNoC). The task graphs of the real applications are generated from the traces of SPLASH-2 and PARSEC [108]. Various benchmarks were used to assess the design of efficient NoC architectures. Some compiler benchmark suites such as MiBench [131] and Livermore [132] were also employed. However, PARSEC and SPLASH were the most commonly used.

7 Conclusion and Future Scope

With the rising need for faster and smaller ICs, mapping and scheduling strategies play an important role in the evolution of three-dimensional NoC architectures. It is obvious that the capabilities of 3D NoC architecture can be fully leveraged by employing effective mapping and scheduling approaches. Each node in 3D is connected to a router and hence routing has a considerable impact on the efficiency of scheduling and mapping. This paper surveys various ongoing mapping and scheduling techniques through an algorithmic viewpoint for regular 3D mesh-based NoCs which satisfy various optimization constraints such as performance, communication cost, temperature, energy consumption, and reliability. We have arranged these works in this paper based on the optimization objectives, simulators and benchmarks adopted, and the approach employed for mapping.
and scheduling. However, there are several challenges and opportunities related to optimization, generalizability, multi-core considerations, reliability, and the application or combination of techniques with more intelligent techniques as discussed below.

- Although different scheduling and mapping strategies claim to optimize different parameters, most of them ignore the increasing time for optimization. Further research can be done on reducing the run time overhead of the proposed algorithms.

The work presented by authors in [87] which is a run-time application mapping framework that can be further extended to investigate support for variable process variations, and may consider a service queue model that includes the wait time of an application. Similarly, 3D mesh structures in [36] can be utilized to yield shorter task completion times, without compromising thermal constraints.

- The growth in power density and heat dissipation makes the 3D-NoC system more easily get overheated and harms the reliability of the system. The router is a potential hotspot here because of the frequent packet routing while running the applications after they are mapped. To meet the temperature criteria, the router microarchitecture and routing algorithm must be adequately developed [110]. This area needs to be further explored and hotspots on NoC routers need to be considered while performing mapping, to explore further thermal optimization on 3D NoC architecture.

- Instead of considering a single factor in scheduling objectives, we need to broaden our research to include multi-objective optimization, which involves optimizing more than one performance indicator at the same time. For instance, DVFS can be applied in [60] along with the thermal-aware scheduling algorithm to further reduce the peak temperature and enhance performance.

- Security is a critical aspect that has not been addressed in any of the works that work on scheduling and mapping of applications. Emerging 3D systems try to incorporate a large number of layers which increases the temperature gradient because only one layer is attached to the heat sink. The temperature standard deviation for the 4-layer 3D chip is roughly 40 times greater than that of the 1-layer 2D device [12]. A Trojan can exploit this flaw by flooding the topmost layer with bogus and/or valid traffic, increasing the likelihood of surpassing the temperature threshold and inducing thermal throttling. Using the algorithms mentioned in this survey paper, we may include secure mapping and scheduling strategies for the upcoming research opportunities.

- Thermal-aware test scheduling has attracted a lot of research interest as this issue has become increasingly important for high-performance testing systems. In order to handle testing scenarios in 3D NoC, researchers developed different mapping and scheduling algorithms for the 3D NoC test scheduling problem. However, an inefficient approach to the NOC testing problem does not utilize the full NOC bandwidth for testing and instead sequentially delivers tests to cores which significantly increases test time and test cost [133]. One solution is that we can employ partitioning of test schedules with interleaving of tests to improve the performance with respect to temperature control and total time consumed for testing the 3D NoC architectures.

- The reliability of three-dimensional systems has yet to be considered in scheduling and mapping strategies. Future work may include a reliability model for the hybrid 2D router (as
proposed in [28] to address TSV yield and failure issues, thereby increasing the quality of service.

- Most of the papers target mesh topology of 3D NoC while designing scheduling and mapping policy. The research on scheduling and mapping on 3D NoC can be extended for other topologies as well. Furthermore, designing policies for a generic 3D NoC topology may also be explored.

- In recent years, the number of ML-based applications has increased dramatically. Machine Learning plays an important role in designing an efficient 3D NoC architecture. Few authors currently explore this field with some limitations [134]. ML models are highly adaptive and these may be explored to find efficient mapping catering to the dynamicity of the system such as temperature, bandwidth, etc. Heterogenous multi-core architectures running multi-threaded programs result in complex task mapping and load balancing. Artificial neural network (ANN) based load balancing techniques may also be combined with the aforementioned mapping and scheduling techniques to further improve the system performance.

References

[1] David Goodwin, Chris Rowen, and Grant Martin. “Configurable Multi-Processor Platforms for Next Generation Embedded Systems”. In: 2007 Asia and South Pacific Design Automation Conference. 2007, pp. 744–746. DOI: 10.1109/ASPDAC.2007.358076

[2] H. Wong. “On the CMOS Device Downsizing, More Moore, More than Moore, and More-than-Moore for More Moore”. In: 2021 IEEE 32nd International Conference on Microelectronics (MIEL). 2021, pp. 9–15. DOI: 10.1109/MIEL52794.2021.9569101

[3] Ahmed Jerraya and Wayne Wolf. Multiprocessor Systems-on-Chips. Jan. 2005. ISBN: 978-0-12-385251-9.

[4] Gordon E. Moore. “Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff. In: IEEE Solid-State Circuits Society Newsletter 11.3 (2006), pp. 33–35. DOI: 10.1109/S-SSC.2006.4785860

[5] Wayne Wolf, Ahmed Amine Jerraya, and Grant Martin. “Multiprocessor System-on-Chip (MPSoC) Technology”. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 27.10 (2008), pp. 1701–1713. DOI: 10.1109/TCAD.2008.923415

[6] William James Dally and Brian Patrick Towles. Principles and Practices of Interconnection Networks. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004. ISBN: 9780080497808.

[7] Konstantinos Tatas et al. Designing 2D and 3D Network-on-Chip Architectures. Springer Publishing Company, Incorporated, 2013. ISBN: 1461442737.

[8] Vasilis F. Pavlidis and Eby G. Friedman. “3-D Topologies for Networks-on-Chip”. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems 15.10 (2007), pp. 1081–1090. DOI: 10.1109/TVLSI.2007.893649

[9] Brett Stanley Feero and Partha Pratim Pande. “Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation”. In: IEEE Transactions on Computers 58.1 (2009), pp. 32–45. DOI: 10.1109/TC.2008.142
[10] Ewerson Carvalho et al. “Evaluation of static and dynamic task mapping algorithms in NoC-based MPSoCs”. In: 2009 International Symposium on System-on-Chip. 2009, pp. 087–090. DOI: 10.1109/S0CC.2009.5335672

[11] Nassima Kadri and Mouloud Koudil. “A Survey on Fault-Tolerant Application Mapping Techniques for Network-on-Chip”. In: Journal of Systems Architecture 92 (Oct. 2018). DOI: 10.1016/j.sysarc.2018.10.001

[12] Amin Sarihi et al. “A Survey on the Security of Wired, Wireless, and 3D Network-on-Chips”. In: IEEE Access 9 (2021), pp. 10762–107656. DOI: 10.1109/ACCESS.2021.3100540

[13] Reetuparna Das et al. “Application-core mapping policies to reduce memory system interference in multi-core systems”. In: 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA). 2013, pp. 107–118. DOI: 10.1109/HPCA.2013.6522311

[14] C. Addo-Quaye. “Thermal-aware mapping and placement for 3-D NoC designs”. In: Proceedings 2005 IEEE International SOC Conference. 2005, pp. 25–28. DOI: 10.1109/S0CC.2005.1554477

[15] O. Hammami et al. “3D IC implementation for MPSOC architectures: Mesh and butterfly based NoC”. In: 2012 4th Asia Symposium on Quality Electronic Design (ASQED). 2012, pp. 155–159. DOI: 10.1109/ASQED.2012.6320492

[16] Amit Kumar Singh et al. “Mapping on multi/many-core systems: Survey of current and emerging trends”. In: 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC). 2013, pp. 1–10. DOI: 10.1145/2463209.2488734

[17] Pradip Kumar Sahu and Santanu Chattopadhyay. “A survey on application mapping strategies for Network-on-Chip design”. In: Journal of Systems Architecture 59.1 (2013), pp. 60–76. ISSN: 1383-7621. DOI: https://doi.org/10.1016/j.sysarc.2012.10.004 URL: https://www.sciencedirect.com/science/article/pii/S1383762112000902

[18] Xu Chuanpei, Guo Rong, and Zhao Tongzhou. “Research on test scheduling of 3D NoC based the strategy of bandwidth division multiplexing and multi-clock”. In: 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI). 2017, pp. 263–268. DOI: 10.1109/ICEMI.2017.8265951

[19] Wilfred Gomes et al. “Ponte Vecchio: A Multi-Tile 3D Stacked Processor for Exascale Computing”. In: 2022 IEEE International Solid- State Circuits Conference (ISSCC). Vol. 65. 2022, pp. 42–44. DOI: 10.1109/ISSCC42614.2022.9731673

[20] Simon Knowles. “Graphcore”. In: 2021 IEEE Hot Chips 33 Symposium (HCS). IEEE. 2021, pp. 1–25.

[21] John Wuu et al. “3D V-Cache: the Implementation of a Hybrid-Bonded 64MB Stacked Cache for a 7nm x86-64 CPU”. In: 2022 IEEE International Solid- State Circuits Conference (ISSCC). Vol. 65. 2022, pp. 428–429. DOI: 10.1109/ISSCC42614.2022.9731565

[22] Thomas Xu et al. “An analysis of designing 2D/3D chip multiprocessor wit different cache architecture”. In: Dec. 2010, pp. 1–6. DOI: 10.1109/NORCHIP.2010.5689433

[23] Mohamad Hairol Jabbar, Dominique Houzet, and Omar Hammami. “Impact of 3D IC on NoC Topologies: A Wire Delay Consideration”. In: 2013 Euromicro Conference on Digital System Design. 2013, pp. 68–72. DOI: 10.1109/DSD.2013.135
[24] Hiroki Matsutani, Michihiro Koibuchi, and Hideharu Amano. “Tightly-Coupled Multi-Layer Topologies for 3-D NoCs”. In: 2007 International Conference on Parallel Processing (ICPP 2007). 2007, pp. 75–75. DOI: 10.1109/ICPP.2007.79.

[25] B Veadesh, S Ravi, and B Venkatapragadeesh. “Design and Analysis of an efficient 3D – Network On – Chip (NoC) Router”. In: 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE). 2018, pp. 1857–1862. DOI: 10.1109/ICRIEECE44171.2018.9009362.

[26] Michael Opoku Agyeman, Ali Ahmadinia, and Nader Bagherzadeh. “Performance and Energy Aware Inhomogeneous 3D Networks-on-Chip Architecture Generation”. In: IEEE Transactions on Parallel and Distributed Systems 27.6 (2016), pp. 1756–1769. DOI: 10.1109/TPDS.2015.2457444.

[27] Jan Joseph et al. “NoCs in Heterogeneous 3D SoCs: Co-Design of Routing Strategies and Microarchitectures”. In: IEEE Access PP (Sept. 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2942129.

[28] Michael Opoku Agyeman et al. “Extending the Performance of Hybrid NoCs beyond the Limitations of Network Heterogeneity”. In: Journal of Low Power Electronics and Applications 7 (Apr. 2017), p. 8. DOI: 10.3390/jlpea7020008.

[29] Zhen Zhang et al. “An inductive-coupling interconnected application-specific 3D NoC design”. In: 2013 IEEE International Symposium on Circuits and Systems (ISCAS). 2013, pp. 550–553. DOI: 10.1109/ISCAS.2013.6571902.

[30] Amir M. Rahmani et al. “Design and Management of High Performance, Reliable, and Thermal-Aware 3D Networks-on-Chip”. In: IET Circuits Devices & Systems 6 (Sept. 2012), pp. 308–321. DOI: 10.1049/iet-cds.2011.0349.

[31] Kostas Siozios, Iraklis Anagnostopoulos, and Dimitrios Soudris. “A High-Level Mapping Algorithm Targeting 3D NoC Architectures with Multiple Vdd”. In: 2010 IEEE Computer Society Annual Symposium on VLSI. 2010, pp. 444–445. DOI: 10.1109/ISVLSI.2010.98.

[32] Jayshree, Gopalakrishnan Seetharaman, and Debadatta Pati. “Energy consumption and performance comparison of DE optimization and PSO-based IP-core mapping technique for 2D and 3D network-on-chip”. In: Semiconductor Science and Technology 36 (Aug. 2021), p. 085015. DOI: 10.1088/1361-6641/ac038c.

[33] Yu Gan, Hong Guo, and Ziheng Zhou. “3D NoC Low-Power Mapping Optimization Based on Improved Genetic Algorithm”. In: Micromachines 12 (Oct. 2021), p. 1217. DOI: 10.3390/mi12101217.

[34] Robert I. Davis and Alan Burns. “A Survey of Hard Real-time Scheduling for Multiprocessor Systems”. In: ACM Comput. Surv. 43.4 (Oct. 2011), 35:1–35:44. ISSN: 0360-0300.

[35] Taotao Zhang et al. “A Traffic Equilibrium Mapping Method with Energy Minimization for 3D NoC-Bus Mesh Architecture”. In: IAENG International Journal of Computer Science 42 (Feb. 2015), pp. 1–7.

[36] Ayhan Demiriz, Hamzeh Ahangari, and Ozcan Ozturk. “Temperature-Aware Core Mapping for Heterogeneous 3D NoC Design Through Constraint Programming”. In: 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). 2020, pp. 312–318. DOI: 10.1109/PDP50117.2020.00054.
[37] Akram Ben Ahmed and Abderazek Ben Abdallah. “LA-XYZ: Low Latency, High Throughput Look-Ahead Routing Algorithm for 3D Network-on-Chip (3D-NoC) Architecture”. In: 2012 IEEE 6th International Symposium on Embedded Multicore SoCs. 2012, pp. 167–174. DOI: 10.1109/MCSoC.2012.24

[38] Akram Ben Ahmed and Abderazek Ben Abdallah. “Architecture and Design of High-Throughput, Low-Latency, and Fault-Tolerant Routing Algorithm for 3D-Network-on-Chip (3D-NoC)”. In: J. Supercomput. 66.3 (Dec. 2013), pp. 1507–1532. ISSN: 0920-8542. DOI: 10.1007/s11227-013-0940-9 URL: https://doi.org/10.1007/s11227-013-0940-9

[39] Parisa Khadem Hamedani et al. “Exploration of Temperature Constraints for Thermal Aware Mapping of 3D Networks on Chip”. In: 2012 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing. 2012, pp. 499–506. DOI: 10.1109/PDP.2012.68

[40] Ismail Akturk and Ozcan Ozturk. “ILP-Based Communication Reduction for Heterogeneous 3D Network-on-Chips”. In: 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. 2013, pp. 514–518. DOI: 10.1109/PDP.2013.83

[41] Parisa Khadem Hamedani et al. “Exploration of Temperature Constraints for Thermal Aware Mapping of 3D Networks on Chip”. In: 2012 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing. 2012, pp. 499–506. DOI: 10.1109/PDP.2012.68

[42] Yigitcan Nalci et al. “ILP Formulation and Heuristic Method for Energy-Aware Application Mapping on 3D-NoCs”. In: J. Supercomput. 77.3 (Mar. 2021), pp. 2667–2680. ISSN: 0920-8542. DOI: 10.1007/s11227-020-03365-0 URL: https://doi.org/10.1007/s11227-020-03365-0

[43] Bing Li et al. “On Runtime Communication and Thermal-Aware Application Mapping and Defragmentation in 3D NoC Systems”. In: IEEE Transactions on Parallel and Distributed Systems 30.12 (2019), pp. 2775–2789. DOI: 10.1109/TPDS.2019.2921542

[44] Yingnan Cui et al. “Thermal-aware task scheduling for 3D-network-on-chip: A Bottom-to-Top scheme”. In: 2014 International Symposium on Integrated Circuits (ISIC). 2014, pp. 224–227. DOI: 10.1109/ISICIR.2014.7029547

[45] Dharm Singh Pranav Wadhwani Naveen Choudhary. “Energy Efficient Mapping in 3D Mesh Communication Architecture for NoC”. In: Global Journal of Computer Science and Technology (2013). ISSN: 0975-4172. URL: https://computerresearch.org/index.php/computer/article/view/233.

[46] Pranav Wadhwani, Naveen Choudhary, and Dharm Singh. “Energy Efficient Mapping in 3D Mesh Communication Architecture for NoC”. In: Global journal of computer science and technology 13 (2013).

[47] Michael Opoku Agyeman, Ali Ahmadinia, and Nader Bagherzadeh. “Performance and Energy Aware Inhomogeneous 3D Networks-on-Chip Architecture Generation”. In: IEEE Transactions on Parallel and Distributed Systems 27.6 (2016), pp. 1756–1769. DOI: 10.1109/TPDS.2015.2457444.

[48] Yingnan Cui, Wei Zhang, and Hao Yu. “Distributed thermal-aware task scheduling for 3D Network-on-Chip”. In: 2012 IEEE 30th International Conference on Computer Design (ICCD). 2012, pp. 494–495. DOI: 10.1109/ICCD.2012.6378690
[49] Ridha Salem, Yahia Salah, and Mohamed Atri. “Decentralized Scheduler for 3D NoC with QoS”. In: International Journal of Computer Science and Information Security, vol. 14 (Dec. 2016), pp. 746–751.

[50] Ting-Hao Tsai and Ya-Shu Chen. “Thermal-Aware Real-Time Task Scheduling for Three-Dimensional Multicore Chip”. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. SAC ’12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 1618–1624. ISBN: 9781450308571. DOI: 10.1145/2245276.2232035 URL: https://doi.org/10.1145/2245276.2232035

[51] Sanghoon Kwak. “Partition-Based Task Mapping for Communication Energy Minimization in 3D Network-on-Chip”. In: Jan. 2021, pp. 43–53. ISBN: 978-981-15-8349-0. DOI: 10.1007/978-981-15-8350-6_4.

[52] Haytham Elmiligi, Fayez Gebali, and M. W. El-Kharashi. “Power-aware Mapping for 3D-NoC Designs Using Genetic Algorithms”. In: Procedia Computer Science 34 (Dec. 2014), pp. 538–543. DOI: 10.1016/j.procs.2014.07.065

[53] Biresh Kumar Joardar et al. “Learning-Based Application-Agnostic 3D NoC Design for Heterogeneous Manycore Systems”. In: IEEE Transactions on Computers 68.6 (2019), pp. 852–866. DOI: 10.1109/TC.2018.2889053

[54] Sourav Das et al. “Optimizing 3D NoC design for energy efficiency: A machine learning approach”. In: Sept. 2015. DOI: 10.1109/ICCAD.2015.7372639

[55] Chiao-Ling Lung et al. “Thermal-aware on-line task allocation for 3D multi-core processor throughput optimization”. In: 2011 Design, Automation Test in Europe. 2011, pp. 1–6. DOI: 10.1109/DATE.2011.5763008

[56] C. Addo-Quaye. “Thermal-aware mapping and placement for 3-D NoC designs”. In: Proceedings 2005 IEEE International SOC Conference. 2005, pp. 25–28. DOI: 10.1109/SOCC.2005.1554447

[57] Jiawen Wang et al. “Latency-aware mapping for 3D NoC using rank-based multi-objective genetic algorithm”. In: 2011 9th IEEE International Conference on ASIC. 2011, pp. 413–416. DOI: 10.1109/ASIC.2011.6157209

[58] Lili Shen et al. “Thermal-aware task mapping for communication energy minimization on 3D NoC”. In: IEICE Electron. Express 14 (2017), p. 20170900.

[59] Gui Feng et al. “MSP based thermal-aware mapping approach for 3D Network-on-Chip under performance constraints”. In: IEICE Electronics Express 13.7 (2016), pp. 20160082–20160082. DOI: 10.1587/elex.13.20160082

[60] Zuomin Zhu et al. “Two-stage thermal-aware scheduling of task graphs on 3D multi-cores exploiting application and architecture characteristics”. In: 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC). 2017, pp. 324–329. DOI: 10.1109/ASPDAC.2017.7858343

[61] C. Addo-Quaye. “Thermal-aware mapping and placement for 3-D NoC designs”. In: Proceedings 2005 IEEE International SOC Conference. 2005, pp. 25–28. DOI: 10.1109/SOCC.2005.1554447

[62] Hanna He, Fang Fang, and Wei Wang. “Improved Simulated Annealing Genetic Algorithm based Low power mapping for 3D NoC”. In: MATEC Web of Conferences 232 (Jan. 2018), p. 02022. DOI: 10.1051/matecconf/201823202022
[63] Gui Feng et al. “A thermal-aware mapping algorithm for 3D Mesh Network-on-Chip architecture”. In: 2013 IEEE 10th International Conference on ASIC. 2013, pp. 1–4. DOI: 10.1109/ASIC.2013.6811834

[64] Bixia Zhang et al. “Thermal and competition aware mapping for 3D network-on-chip”. In: IEICE Electronics Express 9 (Oct. 2012), pp. 1510–1515. DOI: 10.1587/elex.9.1510

[65] Liang Guo et al. “A Novel IP-Core Mapping Algorithm in Reliable 3D Optical Network-on-Chips”. In: Optical Switching and Networking 27 (Aug. 2017). DOI: 10.1016/j.osn.2017.08.001

[66] Haoyuan Ying, Thomas Hollstein, and Klaus Hofmann. “Fast and Optimized Task Allocation Method for Low Vertical Link Density 3-Dimensional Networks-on-Chip Based Many Core Systems”. In: Jan. 2013, pp. 1777–1782. ISBN: 978-1-4673-5071-6. DOI: 10.7873/DATE.2013.357

[67] Jiazheng Li et al. “Bat Algorithm Based Low Power Mapping Methods for 3D Network-on-Chips”. In: NCTCS. 2017.

[68] Kartikeya Bhardwaj and Pravin S. Mane. “C3Map and ARPSO based mapping algorithms for energy-efficient regular 3-D NoC architectures”. In: Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test. 2014, pp. 1–4. DOI: 10.1109/VLSI-DAT.2014.6834909.

[69] Kanchan Manna et al. “Integrated Through-Silicon Via Placement and Application Mapping for 3D Mesh-Based NoC Design”. In: ACM Trans. Embed. Comput. Syst. 16.1 (Nov. 2016). ISSN: 1539-9087. DOI: 10.1145/2968446 URL: https://doi.org/10.1145/2968446

[70] Maamar Bougherara et al. “Efficient Application Mapping onto Three-Dimensional Network-on-Chips Using Multi-Objective Particle Swarm Optimization”. In: June 2019, pp. 654–670. ISBN: 978-3-030-24295-4. DOI: 10.1007/978-3-030-24296-1_53

[71] Cui Huang, Dakun Zhang, and Guozi Song. “A novel mapping algorithm for three-dimensional network on chip based on quantum-behaved particle swarm optimization”. In: Frontiers of Computer Science 11.4 (2017), pp. 622–631. DOI: 10.1007/s11704-016-5196-0.

[72] Cui Huang, Dakun Zhang, and Guozi Song. “Low-power mapping algorithm for three-dimensional network-on-chip based on diversity-controlled quantum-behaved particle swarm optimization”. In: Journal of Algorithms & Computational Technology 10.3 (2016), pp. 176–186. DOI: 10.1177/1748301816649070

[73] Gokul Chandrasekar et al. “Test Scheduling of System-on-Chip Using Dragonfly and Ant Lion Optimization Algorithms”. In: 40.3 (Jan. 2021), pp. 4905–4917. ISSN: 1064-1246. DOI: 10.3233/JIFS-201691 URL: https://doi.org/10.3233/JIFS-201691

[74] Wei Gao, Zhiliang Qian, and Pingqiang Zhou. “Reliability- and Performance-Driven Mapping for Regular 3D NoCs Using a Novel Latency Model and Simulated Allocation”. In: Integr. VLSI J. 65.C (Mar. 2019), pp. 351–361. ISSN: 0167-9260. DOI: 10.1016/j.vlsi.2018.04.012 URL: https://doi.org/10.1016/j.vlsi.2018.04.012

[75] Dongjin Lee et al. “Performance and Thermal Tradeoffs for Energy-Efficient Monolithic 3D Network-on-Chip”. In: ACM Trans. Des. Autom. Electron. Syst. 23.5 (Aug. 2018). ISSN: 1084-4309. DOI: 10.1145/3223046 URL: https://doi.org/10.1145/3223046

38
[76] Amin Mosayyebzadeh, Maziar Mehdizadeh Amiraski, and Shaahin Hessabi. “Thermal and power aware task mapping on 3D Network on Chip”. In: Computers & Electrical Engineering 51 (2016), pp. 157–167. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2015.12.001 URL: https://www.sciencedirect.com/science/article/pii/S0045790615004280

[77] Vivek Chaturvedi et al. “Thermal-aware task scheduling for peak temperature minimization under periodic constraint for 3D-MPSoCs”. In: 2014 25nd IEEE International Symposium on Rapid System Prototyping. 2014, pp. 107–113. DOI: 10.1109/RSP.2014.6966900

[78] Chien-Hui Liao and Charles H.-P. Wen. “Thermal-Constrained Task Scheduling on 3-D Multicore Processors for Throughput-and-Energy Optimization”. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems 23.11 (2015), pp. 2719–2723. DOI: 10.1109/TVLSI.2014.2360802.

[79] Aravindhan Alagarsamy, Lakshminarayanan Gopalakrishnan, and Seok-Bum Ko. “KBMA: A knowledge-based multi-objective application mapping approach for 3D NoC”. In: IET Computers & Digital Techniques 13.4 (2019), pp. 324–334. DOI: https://doi.org/10.1049/iet-cdt.2018.5055 eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-cdt.2018.5055 URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cdt.2018.5055

[80] Lili Shen, Gaizhen Yan, and Jinbao Zhang. “CTTA : A Cluster-Based Thermal-Aware Task Allocation Algorithm for 3 D NoC”. In.

[81] Karthikeyan Appathurai and P. Kumar. “Randomly prioritized buffer-less routing architecture for 3D Network on Chip”. In: Computers & Electrical Engineering 59 (Apr. 2017), pp. 39–50. DOI: 10.1016/j.compeleceng.2017.03.006

[82] Karthikeyan Appathurai and P. Kumar. “GALS implementation of randomly prioritized buffer-less routing architecture for 3D NoC”. In: Cluster Computing 21 (Mar. 2018). DOI: 10.1007/s10586-017-0979-0

[83] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. 1st. USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN: 0201157675.

[84] M. Srinivas and L.M. Patnaik. “Adaptive probabilities of crossover and mutation in genetic algorithms”. In: IEEE Transactions on Systems, Man, and Cybernetics 24.4 (1994), pp. 656–667. DOI: 10.1109/21.286385.

[85] Suleyman Tosun, Ozcan Ozturk, and Meltem Ozen. “An ILP formulation for application mapping onto Network-on-Chips”. In: Nov. 2009, pp. 1–5. DOI: 10.1109/ICAICT.2009.5372524

[86] Xiaohang Wang et al. “Energy Efficient Run-Time Incremental Mapping for 3-D Networks-on-Chip”. In: Journal of Computer Science and Technology 28 (Feb. 2013). DOI: 10.1007/s11390-013-1312-x

[87] Venkata Yaswanth Raparti, Nishit Kapadia, and Sudeep Pasricha. “ARTEMIS: An Aging-Aware Runtime Application Mapping Framework for 3D NoC-Based Chip Multiprocessors”. In: IEEE Transactions on Multi-Scale Computing Systems 3.2 (2017), pp. 72–85. DOI: 10.1109/TMSCS.2017.2686856

[88] J. Sepulveda et al. “3DMIA: a multi-objective artificial immune algorithm for 3D-MPSoC multi-application 3D-NoC mapping”. In: July 2013, pp. 167–168. DOI: 10.1145/2464576.2464659

39
[99] Dong Xiang et al. “Thermal-aware test scheduling for NOC-based 3D integrated circuits”. In: 2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC). 2013, pp. 96–101. DOI:10.1109/VLSI-SoC.2013.6673257

[100] Han Wang et al. “Thermal management via task scheduling for 3D NoC based multi-processor”. In: 2010 International SoC Design Conference. 2010, pp. 440–444. DOI:10.1109/SOCDC.2010.5682875

[98] Y. Zorian. “Test requirements for embedded core-based systems and IEEE P1500”. In: Proceedings International Test Conference 1997. 1997, pp. 191–199. DOI:10.1109/TEST.1997.639613
Kirpan Puttaswamy and Gabriel H. Loh. “Thermal Analysis of a 3D Die-Stacked High-Performance Microprocessor”. In: Proceedings of the 16th ACM Great Lakes Symposium on VLSI. GLSVLSI '06. Philadelphia, PA, USA: Association for Computing Machinery, 2006, pp. 19–24. ISBN: 1595933476. DOI: 10.1145/1127908.1127915 URL: https://doi.org/10.1145/1127908.1127915

Shaobo Liu et al. “Thermal-aware job allocation and scheduling for three dimensional chip multiprocessor”. In: 2010 11th International Symposium on Quality Electronic Design (ISQED). 2010, pp. 390–398. DOI: 10.1109/ISQED.2010.5450547

Chong Sun, Li Shang, and Robert P. Dick. “Three-dimensional multiprocessor system-on-chip thermal optimization”. In: 2007 5th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS). 2007, pp. 117–122.

Eung S. Shin, Vincent J. Mooney, and George F. Riley. “Round-Robin Arbiter Design and Generation”. In: ISSS '02. Kyoto, Japan: Association for Computing Machinery, 2002, pp. 243–248. ISBN: 158415769. DOI: 10.1109/ISVLSI.2010.98 URL: https://doi.org/10.1145/581199.581253

Suyog K Dahule and MA Gaikwad. “Design and simulation of round robin arbiter for NOC architecture”. In: Int. J. Eng. Adv. Technol.(IJEAT) 1.6 (2012).

Avik Bose and Prasun Ghosal. “A low latency energy efficient BFT based 3D NoC design with zone based routing strategy”. In: Journal of Systems Architecture 108 (Sept. 2020), p. 101738. DOI: 10.1016/j.sysarc.2020.101738

Avik Bose, Prasun Ghosal, and Saraju P. Mohanty. “A Low Latency Scalable 3D NoC Using BFT Topology with Table Based Uniform Routing”. In: 2014 IEEE Computer Society Annual Symposium on VLSI. 2014, pp. 136–141. DOI: 10.1109/ISVLSI.2014.51

Bing Li et al. “On runtime communication- and thermal-aware application mapping in 3D NoC”. In: 2017 Eleventh IEEE/ACM International Symposium on Networks-on-Chip (NOCS). 2017, pp. 1–8.

Kostas Siozios, Iraklis Anagnostopoulos, and Dimitrios Soudris. “A High-Level Mapping Algorithm Targeting 3D NoC Architectures with Multiple Vdd”. In: 2010 IEEE Computer Society Annual Symposium on VLSI. 2010, pp. 444–445. DOI: 10.1109/ISVLSI.2010.98

Han Wang et al. “Thermal management via task scheduling for 3D NoC based multi-processor”. In: 2010 International SoC Design Conference. 2010, pp. 440–444. DOI: 10.1109/SDC.2010.5682875

Majid Janidarmian, Ahmad Khademzadeh, and Misagh Tavanpour. “Onyx: A new heuristic bandwidth-constrained mapping of cores onto tile-based Network on Chip”. In: IEICE Electronics Express 6.1 (2009), pp. 1–7.

Suleyman Tosun. “New heuristic algorithms for energy aware application mapping and routing on mesh-based NoCs”. In: Journal of Systems Architecture 57.1 (2011), pp. 69–78.

Srinivasan Murali and Giovanni De Micheli. “Bandwidth-constrained mapping of cores onto NoC architectures”. In: Proceedings design, automation and test in Europe conference and exhibition. Vol. 2. IEEE. 2004, pp. 896–901.

Jingcao Hu and R. Marculescu. “Energy- and performance-aware mapping for regular NoC architectures”. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 24.4 (2005), pp. 551–562. DOI: 10.1109/TCAD.2005.844106
[115] Xiuyi Zhou et al. “Thermal-Aware Task Scheduling for 3D Multicore Processors”. In: *IEEE Transactions on Parallel and Distributed Systems* 21.1 (2010), pp. 60–71. DOI: 10.1109/TPDS.2009.27

[116] Mohammad Arjomand and Hamid Sarbazi-Azad. “Voltage-Frequency Planning for Thermal-Aware, Low-Power Design of Regular 3-D NoCs”. In: 2010 23rd International Conference on VLSI Design. 2010, pp. 57–62. DOI: 10.1109/VLSI.Design.2010.56

[117] Navid Habibi, m salehnamadi m, and Ahmad Khadem-Zadeh. “High-Performance 3D Mesh-Based NOC Architecture Using Node-Layer Clustering”. In: *IETE Journal of Research* (Oct. 2020). DOI: 10.1080/03772063.2020.1830864

[118] Chih-Hao Chao et al. “Traffic- and Thermal-Aware Run-Time Thermal Management Scheme for 3D NoC Systems”. In: 2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip. 2010, pp. 223–230. DOI: 10.1109/NOCs.2010.32

[119] Vincenzo Catania et al. “Cycle-Accurate Network on Chip Simulation with Noxim”. In: 27.1 (Aug. 2016). ISSN: 1049-3301. DOI: 10.1145/2953878 URL: https://doi.org/10.1145/2953878

[120] Nan Jiang et al. “A detailed and flexible cycle-accurate Network-on-Chip simulator”. In: 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). 2013, pp. 86–96. DOI: 10.1109/ISPASS.2013.6557149

[121] Michael Opoku Agyeman, Ali Ahmadiania, and Nader Bagherzadeh. “Performance and Energy Aware Inhomogeneous 3D Networks-on-Chip Architecture Generation”. In: *IEEE Transactions on Parallel and Distributed Systems* 27.6 (2016), pp. 1756–1769. DOI: 10.1109/TPDS.2015.2457444

[122] Lavina Jain et al. “NIRGAM: a simulator for NoC interconnect routing and application modeling”. In: *Design, automation and test in Europe conference*. IEEE. 2007, pp. 16–20.

[123] Nathan Binkert et al. “The Gem5 Simulator”. In: 39.2 (Aug. 2011), pp. 1–7. ISSN: 0163-5964. DOI: 10.1145/2024716.2024718 URL: https://doi.org/10.1145/2024716.2024718

[124] Access Noxim. URL: http://access.ee.ntu.edu.tw/noxim/index.html

[125] Wei Huang et al. “HotSpot: A compact thermal modeling methodology for early-stage VLSI design”. In: *IEEE Transactions on Very Large Scale Integration Systems - VLSI* 14 (May 2006), pp. 501–513. DOI: 10.1109/TVLSI.2006.876103

[126] Christian Bienia et al. “The PARSEC Benchmark Suite: Characterization and Architectural Implications”. In: *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*. PACT ’08. Toronto, Ontario, Canada: Association for Computing Machinery, 2008, pp. 72–81. ISBN: 9781605582825. DOI: 10.1145/1454115.1454128 URL: https://doi.org/10.1145/1454115.1454128

[127] Steven Cameron Woo et al. “SPLASH-2 programs: Characterization and methodological considerations”. English (US). In: *Conference Proceedings - Annual International Symposium on Computer Architecture, ISCA*. Conference Proceedings - Annual International Symposium on Computer Architecture, ISCA. Proceedings of the 1995 22nd Annual International Symposium on Computer Architecture ; Conference date: 22-06-1995 Through 24-06-1995. Jan. 1995, pp. 24–36. ISBN: 0780330005
[128] Randy Morris, Avinash Karanth Kodi, and Ahmed Louri. “3D-NoC: Reconfigurable 3D photonic on-chip interconnect for multicores”. In: 2012 IEEE 30th International Conference on Computer Design (ICCD). 2012, pp. 413–418. DOI: 10.1109/ICCD.2012.6378672

[129] Thomas Canhao Xu, Pasi Liljeberg, and Hannu Tenhunen. “A study of Through Silicon Via impact to 3D Network-on-Chip design”. In: 2010 International Conference on Electronics and Information Engineering. Vol. 1. 2010, pp. V1-333-V1-337. DOI: 10.1109/ICEIE.2010.5559865

[130] Sourav Das et al. “Design-Space Exploration and Optimization of an Energy-Efficient and Reliable 3-D Small-World Network-on-Chip”. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems PP (Aug. 2016). DOI: 10.1109/TCAD.2016.2604288

[131] M.R. Guthaus et al. “MiBench: A free, commercially representative embedded benchmark suite”. In: Proceedings of the Fourth Annual IEEE International Workshop on Workload Characterization. WWC-4 (Cat. No.01EX538). 2001, pp. 3–14. DOI: 10.1109/WWC.2001.990739

[132] R. D. Hornung. Livermore Compiler Analysis Loop Suite, Version 00. Mar. 2013. URL: https://www.osti.gov/biblio/1231669

[133] Érika Cota, Luigi Carro, and Marcelo Lubaszewski. “Reusing an On-Chip Network for the Test of Core-Based Systems”. In: 9.4 (Oct. 2004), pp. 471–499. ISSN: 1084-4309. DOI: 10.1145/1027084.1027088 URL: https://doi.org/10.1145/1027084.1027088

[134] Sourav Das et al. “Energy-efficient and reliable 3D Network-on-Chip (NoC): Architectures and optimization algorithms”. In: 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). 2016, pp. 1–6. DOI: 10.1145/2966986.2980096