2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)

Temperature-Aware Core Mapping for Heterogeneous 3D NoC Design Through Constraint Programming

Ayhan Demiriz
Gebze Technical University,
Gebze, Kocaeli, 41400, TURKEY
Email: ademiriz@gmail.com

Hamzeh Ahangari
Bilkent University,
Ankara, 06800, TURKEY
Email: hamzeh@bilkent.edu.tr

Ozcan Ozturk
Bilkent University,
Ankara, 06800, TURKEY
Email: ozturk@cs.bilkent.edu.tr

***Abstract*—In the context of Network-on-Chip (NoC) based Chip Multiprocessor (CMP) design, core mapping for application specific systems is a challenging problem. In such designs, various decisions have to be made that affect performance and power consumption. Moreover, in emerging 3D NoC systems, by intensification of cooling issues, temperature constraints on hot-spots are added, and problem becomes more complicated. In this paper, an earlier Constraint Programming (CP) methodology for heterogeneous 2D NoC design is extended to 3D model, while critical temperature constraints are accounted. In a single-stage, our approach can choose core types from a set of low, medium and high power, and assign them to appropriate places on the mesh which minimizes the overall computation time and communication cost while satisfying the temperature constraints. To achieve our objective, in addition to cores placement problem, tasks should also be scheduled on corresponding cores with matching performance levels to minimize the overall completion time (makespan). Experimental results show that task completion times are more dependent on the mesh structure for our benchmark data. 3D mesh structures may yield shorter task completion times, without compromising thermal constraints. On the other hand, restricting the peak temperature naturally requires the usage of low-performance computing elements which inherently may delay the processing time.**

Keywords: Network-on-Chip, 3D Integration, Heterogeneous Core Mapping, Task Scheduling, Constraint Programming.

I. Introduction

Increasing the number of processing elements inside a single Chip Multi-Processor (CMP) integrated-circuit (IC) is a current road-map in semiconductor technology. As the number of cores is raised to several dozen, traditional shared bus would not be a practical solution for interconnecting all cores. Therefore, communication bottleneck is resolved with new interconnection paradigms introduced by Network-on-Chips (NoCs). NoC has become an emerging trend in many-core chip multiprocessor design to tackle limitations of traditional communication mechanisms. Various NoC topologies bring flexibility and performance to communication among cores. Along with NoC, three dimensional (3D) integration is another modern trend to increase the transistor density on chip area. Reducing interconnection delay between cores and/or memories, by allowing vertical links, is another major benefit of 3D die stacking.

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 interconnect capacitance. However, despite all these benefits, a critical dilemma is intensified in higher integration levels. As device density increases, power density increases too and as a result, thermal management and required cooling solution become more challenging. Because of less interconnect capacitance, 3D NoC normally dissipates lower thermal power than an equivalent NoC implemented by multiple packages of 2D ICs . Nevertheless, due to more power density and less direct contact area exposed to ambient air per core, transferring generated thermal energy to the ambient air is more difficult in 3D stack die, in comparison to multiple 2D NoC chips.

In the context of application specific 3D Network-on-Chip systems, core mapping; which means placement of cores inside optimal available space of 3D chip; is one of the challenging problems in the domain of 3D NoC. In this paper, we aim to face this problem from a constraint programming (CP) perspective by a single-stage solution. Given a Communication Task Graph (CTG) and subsequent task assignments for the cores, heterogeneous CPU cores are allocated to the best possible places on the chip in order to minimize the overall communication cost among cores. Concurrently, the application scheduling stage is run to determine the optimum core types from a list of technological alternatives and to minimize the makespan, i.e. time to complete all computation tasks on CTG. Moreover, selection of core type has to satisfy thermal limitations. It means that in worse case, none of cores are allowed to go beyond specified temperature limit. If such adverse situation happens, lifetime of IC is greatly reduced, or even it may damage other nearby units inside system. Improving technology makes the ICs vulnerable to thermal problems due to the increase in power density. This causes an increase in leakage power dissipation and electro-migration which contribute to further higher temperatures [1]. Heterogeneous designs may involve optimization problems that have

2377-5750/20/$31.00 ©2020 IEEE
DOI 10.1109/PDP50117.2020.00054

conflicting terms in their objective functions. To facilitate solutions for the heterogeneous designs, as we will see in Section III, constraint programming formulation and objective functions are introduced and then solved by a commercial CP solver (IBM CPLEX/CP SOLVER).

The main contribution of this paper, in comparison to our past works [2]–[4], is extension of core selection for application specific 2D NoCs, to 3D NoC designs. Temperature constraints and heat transfer formulations are embedded in CP model to provide a static thermal management scheme. The remainder of paper is organized as follows: In Section 2, some related literature is reviewed. Thenceforth, CP formulation of the proposed model is presented in Section 3. Experimental results on real benchmarks are given in Section 4. Finally, we conclude our paper in Section 5.

II. Related Work

In recent years, there have been several works published that study the optimal core mapping and application scheduling problems for heterogeneous NoC architecture in different levels. In [5], authors proposed a comprehensive two-stage NoC synthesis model by utilizing the Mixed-Integer Programming (MIP). In the first stage, an energy efficient system-level floor-planning is achieved through MIP. The second stage is conducted for a detailed routing functionality. At stage two, placement of routers is optimized to enable the traffic flow. The MIP model is very complicated in [5], and it often does not return a solution within the run-time limits. Therefore, a clustering-based heuristic is proposed to address the complexity issue of the second stage. It should be noted that if a certain level of the problem abstraction is not applied appropriately in the MIP models, it is very likely that the MIP models will not able to return a solution within the run-time limits, due to complexity issues.

A two-stage solution to core mapping and application scheduling problems was also proposed in [6]. The solution is reached by running iteratively these two consecutive stages (master and sub-problems). In each iteration, a new cut was introduced to the master problem in order to get closer to the optimal solution, and satisfy the feasibility of scheduling. In [6], the master problem (core mapping) is modeled by integer programming, and sub-problem (scheduling) is modeled by CP. Since there are no task deadlines in our model, it is always feasible to find a solution to the scheduling problem in our case. On the other hand, our scheduling model is finer-grained than the one proposed in [6]. [7] proposes a task scheduling approach that uses statically formed temperature profiles of tasks for mapping them to corresponding cores. Authors in [8] and [9] propose a dynamic approach for task allocation on a homogeneous NoC platform. The objective is to minimize communication cost of application. The work in [10] introduces a constructive heuristic for lowering peak temperature and maintaining thermal variance with controlled task completion time degradation.

[11] proposes a heuristic framework with delay insertion, depending on predicted temperatures, based on actual task durations. Delay is inserted when the temperature limit is exceeded, while a task is being processed. On the other hand, [12] proposes a SVM-based prediction method for temperature, to dynamically schedule the tasks. A heuristic topology synthesis approach is proposed in [13], which includes application clustering to assign cores to specific routers, topology construction to find a routing path for all flows, in addition to link insertion to produce solution topology by interconnecting the routers. Maximum delay and maximum number of links are considered as constraints, while authors claim to improve power consumption and area overhead. In [14], authors propose a heuristic to determine the locations of components, routers and vertical links in 3D NoCs, with five design steps. Method is based on separation of intra-layer and inter-layer communications. Authors showed that the advantage of this method is that this form of the problem can be solved with well-known methods.

A heterogeneous 2D NoC design is proposed in [15], by implementing core mapping as a 2D-packing problem, using a heuristic solution for the underlying optimization problem. Power usage has also been taken into consideration for the scheduling phase. [1] compares both ILP and meta-heuristics methods for a regular 2D mesh-based thermal-aware NoC platform. It proposes a design-time mapping strategy, by using particle swarm optimization based technique.

The main point of difference of this work in comparison to previous works, is the methodology which is used to tackle the problem. Because of clarity and understandability, we find Constraint Programming (CP) a suitable modeling for the problem. In comparison to our own previous works [2]–[4], in this work, the three dimensional modeling, and required thermal constraints have been added to the problem.

III. Proposed Optimization Model

A. Basic assumptions

We assume that a set of Processing Elements (PEs) are arranged inside a 3D mesh structure of size = L x W x H. We limit the height (H) of 3D architecture to 2 or 3. Length (L) and Width (W) are also limited to 3 or 4. Heterogeneous cores are selected from a set of three hypothetical PE cores: Type-H which is high-performance, Type-M which is mid-performance, and Type-L which is low-performance. Each type has different area, performance and power consumption. We assume normalized numbers as listed in Table I. However, these are just some typical values to show how our model results vary in running benchmarks. In temperature calculations, the power of Type-M core is assumed to be 10 Watt.

Core Type	Area Coef.	Speed Coef.	Power Coef.
Type-H	2	1.4	1.8
Type-M	1	1	1
Type-L	0.5	0.7	0.2

TABLE I
Core types

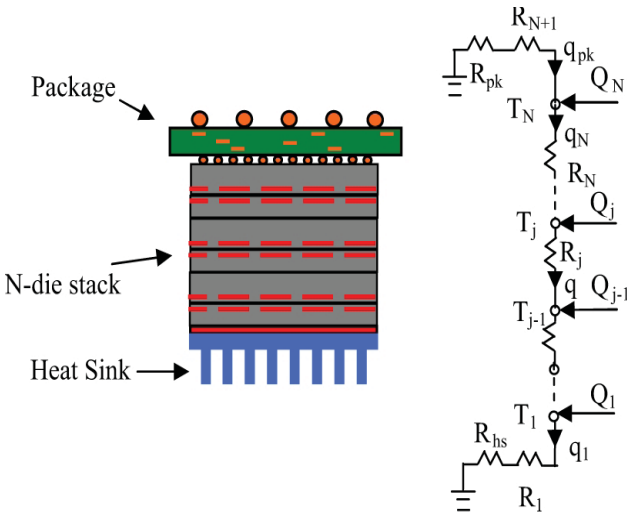

Fig. 1. One dimensional heat transfer model [16]

The optimization solver i.e. CP solver determines on the location of cores to minimize communication cost. We assume that for a specific application, the communication requirement between cores is already known. Communication cost is estimated based on the 3D Manhattan distance between the nodes, as well as the communication intensity. In 3D stacking, vertical communications are performed through TSVs and has to be treated differently. The inter-layer vertical links are shorter and then faster than horizontal intra-layer links. Therefore, we consider less communication cost for vertical links. This can be captured by parameter ρ. For instance, ρ can be taken equal to 0.2 as in [15] with a conservative estimation.

$$CommCost = CommCostH + \rho * CommCostV \quad (1)$$

B. Heat Transfer Model and Related Formulations

Comprehensive heat transfer modeling in stacked 3D die can be a complicated problem, which requires complex system of differential equations to be solved [17]. Heat is generated by any working component, in any layer inside 3D die. After that, the generated and accumulated heat energy flows toward package boundaries, and it is dissipated to ambient air. This may happen mainly through the top side of package, where contact area to air is larger. Possibly a heatsink is connected upon package top side as well. Inside IC, heat flux can flow in any direction depending on temperature difference, from hotter to cooler points, vertically to above and below layers, or horizontally inside the same layer.

Since the layout of any VLSI core is a flat and thin plate, a core has by far greater contact area with cores directly in upper and lower layers, than cores in the same layer. Consequently, inside a 3D IC, the major part of heat flows vertically to above and below layers, not horizontally in the same layer. Thus, according to this argument, several studies like [16], [18] suggested a simplified one-dimensional (1D) heat transfer model, instead of a multi-dimensional complicated model. Some more complex models count the heat capacitance of materials for time domain formulation. It means materials conserve heat in a time and release it in another time, somehow similar to behavior of an electric capacitor in electric circuits. However, in this work, we assume the steady-state model without considering such time domain formulation.

Based on single dimensional heat transfer modeling, Ankur [16] developed an analytical model for heat transfer, or equivalently temperature distribution, in multi-source 3D stack. Such model can be employed to find or predict thermal hotspots in 3D IC, and then apply any thermal management scheme. According to the model, as depicted in Fig 1, thermal resistance network is composed of N vertical heat sources and N+3 thermal resistors. R_{hs} and R_{pk} are thermal resistance of heat sink and package respectively. R_1 is the thermal resistance between bottom heat sources and heat sink. R_{N+1} is thermal resistance between top heat sources and package. R_i represent thermal resistance between internal heat sources. In general, between each two vertical nodes, there are several types of material, namely substrate and interdie micropad layers. However, a single resistor R represents equivalent summation of thermal resistances of all such different materials. Although core area may affect temperature distribution and thermal resistance values, we neglect such parameter. The generated heat at node i is injected to the network and has been shown by Q_i. T_i represents temperature at node i. Heat currents passing throw thermal resistances are shown by q_i. Temperature of ambient air which package and heatsink are in direct contact with it are assumed to be fixed, equal to 20°C.

From a physical perspective, heat generated at each node traverses all other vertical nodes to reach heatsink or package, whereby can be dissipated to air. This means that temperature at each node is obviously affected by generated heat at other nodes. As mentioned in assumption section, this work is limited to 3D stack die with two and three number of layers.

Temperature at each point is calculated by below formulas. First equation states that heat flow magnitude is determined by temperature difference. Second equation states that in steady-state, at each point, summation of inward heat flows is equal to summation of outward heat flows.

Between each two cores:

$$q = \frac{\triangle T}{R} \quad (2)$$

At each core:

$$\sum q + Q = 0 \quad (3)$$

The hypothetical values taken for this work are listed in Table II [16].

C. Underlying CP Model

We provide underlying CP model in this section. CP is primarily used for constraint satisfaction problems. In other words, the main purpose of using CP is to find a feasible solution as an intersection of artificial intelligence (AI) and operations research (OR). It utilizes powerful search algorithms

R_{hs}	2 K/W
R_{pk}	20 K/W
R_i	1 K/W
Q_{high}	18 W
Q_{med}	10 W
Q_{low}	2 W
Ambient temp.	20°C
Max allowed temp.	100°C

TABLE II
TYPICAL VALUES USED IN CALCULATIONS [16]

from AI with a combination of OR techniques. We can also introduce objective function in CP models to either minimize or maximize depending on the underlying problem. The problem definition of our CP model is given as combination of Sets, Parameters, Decision Variables, Decision Expressions (i.e. function of decision variables), Objective Function, and finally Constraints in this section. CP technology allows us to define a comprehensive model easily, with powerful constructs. Heat transfer model is represented in decision expressions which are functions of decision variables and model parameters.

Sets

$\mathscr{T}$, Set of Tasks
$\mathscr{C}$, Set of Cores
$\mathscr{L}$, Set of Links where task graph is embedded and provided in benchmark set

Parameters

M, Number of PE (CPU) types available
S, Layer Size ($L \times W$) (Number of Cores in a layer)
H, Number of Layers (Height of 3D architecture)
T, Maximum Allowable Core Temperature
T_{amb}, Ambient Temperature
R_{pk}, Package Resistance
R_{hs}, Heat Sink Resistance
R, PE (Core) Resistance
$XYZCost_{ij}$, Communication cost between two cores (in number of hubs) where $i,\ j \in 1, \ldots, |\mathscr{C}|$
Υ_i, the corresponding PE ID (number) where a task should be performed, provided in benchmark set, $i \in 1, \ldots, |\mathscr{T}|$
D_i, Duration of Tasks in Clock Cycles where $i \in 1, \ldots, |\mathscr{T}|$, provided in benchmark set
Ω_i, Communication cost between two consecutive tasks on a task graph where $i \in \mathscr{L}$

Decision Variables:

α_{ij} Binary Variable for PE Type decision where $i \in 1, \ldots, |\mathscr{C}|,\ j \in 1, \ldots, M$
γ_i, Job start and end times (interval variables in CP formulation) where $i \in 1, \ldots, |\mathscr{T}|$
β_i, Permutation variable for core placement decision where $i \in 1, \ldots, |\mathscr{C}|$ and $1 \leq \beta_i \leq |\mathscr{C}|$

Decision Expressions

$Q_{ij} = 2.5 * \alpha_{i1} + 5.1 * \alpha_{i2} + 10 * \alpha_{i3}$ where $i \in 1, \ldots, |\mathscr{C}|$
$\theta_{ij} = (R_{hs} + R) * \sum_{k=2}^{j} Q_{ik} + \sum_{k=2}^{j} R * \sum_{l=k}^{H} Q_{il}$ where $i \in 1, \ldots, S$ and $j \in 2, \ldots, H-1$
$\theta_{i1} = \frac{\theta_{i2} - R * Q_{i1} * (R_{hs} + R)}{R_{hs}}$ where $i \in 1, \ldots, S$
$\theta_{iH} = \frac{(\theta_{i(H-1)} + R * Q_{iH}) * (R_{pk} + R)}{R_p k + 2 * R}$ where $i \in 1, \ldots, S$
$\tau_{ij} = \theta_{ij} + T_{amb}$ where $i \in 1, \ldots, S$ and $j \in 1, \ldots, H$
$\omega_i = (\Omega_i + 3 * \lceil \Omega_i / 31 \rceil) * XYZCost_{\beta_{\Upsilon_{i_1}} \beta_{\Upsilon_{i_2}}}$, where $i \in \mathscr{L}$

Objective function:

$$minimize \max_{i \in 1, \ldots, |\mathscr{T}|} endOf(\gamma_i) \tag{4}$$

Constraints:

forall i :
$$sizeOf(\gamma_i) = D_i * (1.4 * \alpha_{\beta_{\Upsilon_i} 1} + \alpha_{\beta_{\Upsilon_i} 2} + 0.7 * \alpha_{\beta_{\Upsilon_i} 3}),$$
$$i \in 1, \ldots, |\mathscr{T}| \tag{5}$$

$$\text{forall i: } \sum_{j=1}^{M} \alpha_{ij} = 1, \quad i \in 1, \ldots, |\mathscr{T}| \tag{6}$$

$$\max_{i \in 1, \ldots, S,\ j \in 1, \ldots, H} \tau_{ij} \leq T \tag{7}$$

$$\text{allDifferent}(\beta) \tag{8}$$

$$\text{forall i: endBeforeStart}(\gamma_{i_1}, \gamma_{i_2}, \omega_i), \quad i \in \mathscr{L} \tag{9}$$

Note that some of the constraint programming statements such as allDifferent, forall and endBeforeStart are used as in OPL syntax. Notice also that execution time of each task is according to the assigned PE type (constraint 5). For each core, a PE type should be assigned (constraint 6). The thermal constraint 7 is satisfied by realizations of all decision expressions except ω_i. Moreover, those decision expressions are all dependent to each other. The constraint 8 simply maps (assigns) each PE to the best corresponding core.

IV. EXPERIMENTAL RESULTS

We have employed benchmark datasets of real applications to evaluate the mapping and scheduling algorithms, in this section. Multi-Constraint System-Level (MCSL) benchmark suite [19] provides a set of real applications, which each application includes multiple tasks, and traffic data patterns between these tasks. MCSL benchmark records the data traffic for different mesh network sizes, and measures the execution time for each task in the application. Most of the architectural settings are borrowed from [2], while exceptions are specified as needed. Results from heterogeneous architectures are presented in this section. The CP models are implemented using IBM CPLEX Studio, which is available free of charge

TABLE III
MCSL BENCHMARK SUITE APPLICATIONS

Application	Number of Tasks	Number of Comm. Links
R-S code encoder	248	328
R-S code decoder	278	390
ROBOT	88	131
SPEC95 FPPPP	334	1145
SPARSE	96	67
H.264 video decoder	2311	3461

TABLE IV
SUMMARY OF GENERAL EXPERIMENTAL SETTINGS

Experiment Set	T_{amb}	R_{pk}	R_{hs}	R
First	26.70	100,000	4	1.33
Second	25	20	2	1.3

to the academicians at IBM Academic Initiative web site. Interested readers can access a representative CP model file at https://tinyurl.com/u5mz84n.

Six datasets are used from MCSL benchmark suite in this study, as in our previous work [2]. Table III shows the applications provided by MCSL, which are used as data sets of our mapping and scheduling algorithms. Table III shows also the number of tasks of each application, as well as the number of communication edges. Two sets of experiments are conducted for each data set. Basically, two sets of heat related parameter settings are used in this paper, as shown in Table IV. 2D and 3D mesh structures are compared in our studies by analyzing 6×6, $3 \times 6 \times 2$, $4 \times 3 \times 3$, $3 \times 3 \times 4$, 8×8, $4 \times 8 \times 2$, and $4 \times 4 \times 4$ cases. The last digit represents number of layers. Therefore, in this paper, the sizes of mesh structures are 36-core and 64-core. 2D cases are only 6×6 and 8×8. The parameter ρ for communication cost, in Equation 1, is set to 1.

Tables V-XVI report task completion times under varying temperature and architecture for each data set. For brevity, architecture types are shown without $\times$ like 66 instead of 6×6. The shortest completion times are shown in **boldface** type. Recall that CP models are run under time limitations without seeking optimality. In other words, CP returns the best solution by the end of runtime for each experiment. Note that CP runtime and task completion times reported in Tables

TABLE V
TASK COMPLETION TIMES IN FIRST SET OF EXPERIMENTS FOR R-S CODE ENCODER

	Architecture						
T	66	362	433	334	88	482	444
90° C	1734	1894	1785	NoSol	1737	1961	NoSol
100° C	1741	1873	1734	NoSol	**1681**	1953	2046
115° C	1745	1813	1741	1733	1702	1954	1718
125° C	1742	1813	1721	1734	1694	1920	1742

TABLE VI
TASK COMPLETION TIMES IN SECOND SET OF EXPERIMENTS FOR R-S CODE ENCODER

	Architecture						
T	66	362	433	334	88	482	444
90° C	1745	1945	NoSol	NoSol	1702	1838	NoSol
100° C	1745	1864	1709	NoSol	1694	1817	NoSol
115° C	1745	1864	1806	NoSol	1702	1826	1913
125° C	1745	1864	**1674**	NoSol	1694	1966	1942

TABLE VII
TASK COMPLETION TIMES IN FIRST SET OF EXPERIMENTS FOR R-S CODE DECODER

	Architecture						
T	66	362	433	334	88	482	444
90° C	2712	2733	**2683**	NoSol	2741	2754	NoSol
100° C	2713	2728	2684	NoSol	2743	2758	NoSol
115° C	2706	2728	2684	2694	2736	2769	2699
125° C	2706	2728	2684	2694	2734	2763	2702

TABLE VIII
TASK COMPLETION TIMES IN SECOND SET OF EXPERIMENTS FOR R-S CODE DECODER

	Architecture						
T	66	362	433	334	88	482	444
90° C	2706	2732	NoSol	NoSol	2735	2759	NoSol
100° C	2706	2728	2692	NoSol	2735	2763	NoSol
115° C	2706	2731	**2690**	NoSol	2735	2771	NoSol
125° C	2706	2731	2692	NoSol	2735	2767	NoSol

TABLE IX
TASK COMPLETION TIMES IN FIRST SET OF EXPERIMENTS FOR ROBOT

	Architecture						
T	66	362	433	334	88	482	444
90° C	91479	91423	**91337**	NoSol	91479	91431	NoSol
100° C	91479	91423	91337	91479	91479	91431	91479
115° C	91479	91423	91337	91479	91479	91431	91479
125° C	91479	91423	91337	91479	91479	91431	91479

TABLE X
TASK COMPLETION TIMES IN SECOND SET OF EXPERIMENTS FOR ROBOT

	Architecture						
T	66	362	433	334	88	482	444
90° C	91479	91423	**91337**	91479	91479	91431	NoSol
100° C	91479	91423	91337	91479	91479	91431	NoSol
115° C	91479	91423	91479	91479	91479	91479	91479
125° C	91479	91423	91479	91479	91479	91479	91479

TABLE XI
TASK COMPLETION TIMES IN FIRST SET OF EXPERIMENTS FOR SPEC95 FPPPP

	Architecture						
T	66	362	433	334	88	482	444
90° C	75040	75246	**74902**	NoSol	74988	75449	NoSol
100° C	75040	75138	74902	NoSol	74988	75450	NoSol
115° C	75040	75278	74902	75040	74988	75408	74988
125° C	75040	75259	74902	75040	74988	75334	74988

TABLE XII
TASK COMPLETION TIMES IN SECOND SET OF EXPERIMENTS FOR SPEC95 FPPPP

	Architecture						
T	66	362	433	334	88	482	444
90° C	75040	75259	NoSol	NoSol	74988	75408	NoSol
100° C	75040	75259	**74902**	NoSol	74988	75334	NoSol
115° C	75040	75244	74902	NoSol	74988	75334	NoSol
125° C	75040	75211	74902	NoSol	74988	75305	NoSol

TABLE XIII
TASK COMPLETION TIMES IN FIRST SET OF EXPERIMENTS FOR SPARSE

	Architecture						
T	66	362	433	334	88	482	444
90° C	19696	19696	19240	NoSol	19448	**19170**	NoSol
100° C	19696	19696	19240	NoSol	19448	19170	NoSol
115° C	19696	19696	19240	19696	19448	19170	19448
125° C	19696	19696	19240	19696	19448	19170	19448

TABLE XIV
TASK COMPLETION TIMES IN SECOND SET OF EXPERIMENTS FOR SPARSE

	Architecture						
T	66	362	433	334	88	482	444
90° C	19696	19696	NoSol	NoSol	19448	**19170**	NoSol
100° C	19696	19696	19240	NoSol	19448	19170	NoSol
115° C	19696	19696	19240	NoSol	19448	19170	NoSol
125° C	19696	19696	19240	NoSol	19448	19170	NoSol

TABLE XV
TASK COMPLETION TIMES IN FIRST SET OF EXPERIMENTS FOR H.264 VIDEO DECODER

	Architecture						
T	66	362	433	334	88	482	444
90° C	18663250	18662910	18662760	NoSol	18662690	18662360	NoSol
100° C	18663250	18662910	18662760	NoSol	18662690	18663170	NoSol
115° C	18663250	18662910	**18662570**	18662940	18662690	18663840	18662590
125° C	18663250	18662910	18662570	18662940	18662690	18663840	18662590

TABLE XVI
TASK COMPLETION TIMES IN SECOND SET OF EXPERIMENTS FOR H.264 VIDEO DECODER

	Architecture						
T	66	362	433	334	88	482	444
90° C	18663250	18662913	NoSol	NoSol	18662690	18663843	NoSol
100° C	18663250	18662913	**18662568**	NoSol	18662690	18663843	NoSol
115° C	18663250	18662913	18662568	NoSol	18662690	18663843	NoSol
125° C	18663250	18662913	18662568	NoSol	18662690	18663843	NoSol

V-XVI are totally two different concepts. CP runtime means that the upper time limit that the solver can find a solution. The latter one is makespan of all the tasks for the 3DNoC.

Intuitively, when temperature limit is increased, one may expect to have shorter task completion time due to the flexibility of using higher-end (TYPE-H) cores. We can see some results in Tables V-XVI supporting this claim, especially in 3D architectures. However, there are some counter-intuitive results too. This is due to the fact that having a harder constraint, such as lower temperature constraints, certainly helps reducing the search space, and then improves the quality of solution, meaning lower task completion time.

We also note that generally speaking, 3D mesh structures perform better than 2D ones. Overall, the best structure in our experiments has 36 cores with 3D mesh of size $4 \times 3 \times 3$.

V. CONCLUSION

In this work we proposed a constraint programming (CP) based model to solve the problem of thermal-aware optimal core mapping and application scheduling for application specific heterogeneous 3D Network-on-Chip architectures. We provide a static thermal management scheme, by applying a thermal-aware core selection approach, to assure that temperature of all processing nodes will not pass predetermined peak limits. The major advantages of such CP based model for designing 3D NoC architectures are clarity, and understandability of model. The model has been applied to various real benchmark data sets successfully. The peak temperature limit varies between 90° C and 125° C. The results show that 3D mesh structures may yield shorter task completion times, without compromising thermal constraints.

REFERENCES

[1] K. Manna, P. Mukherjee, S. Chattopadhyay, and I. Sengupta, "Thermal-aware application mapping strategy for network-on-chip based system design," *IEEE Transactions on Computers*, vol. 67, no. 4, pp. 528–542, April 2018.

[2] A. Demiriz, N. Bagherzadeh, and A. Alhussein, "Using constraint programming for the design of network-on-chip architectures," *Computing*, pp. 1–14, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00607-013-0359-4

[3] A. Demiriz and N. Bagherzadeh, "On heterogeneous network-on-chip design based on constraint programming," in *Proceedings of the Sixth International Workshop on Network on Chip Architectures*, ser. NoCArc '13. New York, NY, USA: ACM, 2013, pp. 29–34. [Online]. Available: http://doi.acm.org/10.1145/2536522.2536528

[4] A. Demiriz, N. Bagherzadeh, and O. Ozturk, "Voltage island based heterogeneous noc design through constraint programming," *Computers & Electrical Engineering*, vol. 40, no. 8, pp. 307 – 316, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790614002183

[5] K. Srinivasan, K. S. Chatha, and G. Konjevod, "Linear-programming-based techniques for synthesis of network-on-chip architectures," *IEEE Trans. VLSI Syst.*, vol. 14, no. 4, pp. 407–420, 2006. [Online]. Available: https://doi.org/10.1109/TVLSI.2006.871762

[6] M. Ruggiero, D. Bertozzi, L. Benini, M. Milano, and A. Andrei, "Reducing the abstraction and optimality gaps in the allocation and scheduling for variable voltage/frequency mpsoc platforms," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 28, no. 3, pp. 378–391, 2009. [Online]. Available: https://doi.org/10.1109/TCAD.2009.2013536

[7] S. Cao, Z. Salcic, Y. Ding, Z. Li, S. Wei, and X. Zhao, "Temperature-aware task scheduling heuristics on network-on-chips," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 2603–2606.

[8] C. Chou and R. Marculescu, "Run-time task allocation considering user behavior in embedded multiprocessor networks-on-chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 1, pp. 78–91, Jan 2010.

[9] C.-L. Chou and R. Marculescu, "User-aware dynamic task allocation in networks-on-chip," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '08. New York, NY, USA: ACM, 2008, pp. 1232–1237. [Online]. Available: http://doi.acm.org/10.1145/1403375.1403675

[10] P. K. Sahu, K. Manna, T. Shah, and S. Chattopadhyay, "Article: Thermal uniformity-aware application mapping for network-on-chip design," *International Journal of Computer Applications*, vol. 99, no. 3, pp. 8–22, August 2014.

[11] T. Chantem, X. S. Hu, and R. P. Dick, "Temperature-aware scheduling and assignment for hard real-time applications on mpsocs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 10, pp. 1884–1897, Oct 2011.

[12] B. Yun, K. G. Shin, and S. Wang, "Predicting thermal behavior for temperature management in time-critical multicore systems," in *2013 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2013, pp. 185–194.

[13] F. Vardi, A. Khadem-Zadeh, and M. Reshadi, "A heuristic clustering approach to use case-aware application-specific network-on-chip synthesis," *The Journal of Supercomputing*, vol. 73, no. 5, pp. 2098–2129, May 2017. [Online]. Available: https://doi.org/10.1007/s11227-016-1905-6

[14] J. M. Joseph, D. Ermel, L. Bamberg, A. G. Ortiz, and T. Pionteck, "System-level optimization of network-on-chips for heterogeneous 3d system-on-chips," *ArXiv*, vol. abs/1909.13807, 2019.

[15] I. Akturk and O. Ozturk, "ILP-based communication reduction for heterogeneous 3d network-on-chips," in *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. IEEE, feb 2013.

[16] A. Jain, R. E. Jones, R. Chatterjee, and S. Pozder, "Analytical and numerical modeling of the thermal performance of three-dimensional integrated circuits," *IEEE Transactions on Components and Packaging Technologies*, vol. 33, no. 1, pp. 56–63, March 2010.

[17] E. Kreyszig, *Advanced Engineering Mathematics*. John Wiley & Sons, 2010. [Online]. Available: https://books.google.co.in/books?id=UnN8DpXI74EC

[18] K. Chen, E. Chang, H. Li, and A. A. Wu, "Rc-based temperature prediction scheme for proactive dynamic thermal management in throttle-based 3d nocs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 206–218, 2015. [Online]. Available: https://doi.org/10.1109/TPDS.2014.2308206

[19] W. Liu, J. Xu, X. Wu, Y. Ye, X. Wang, W. Zhang, M. Nikdast, and Z. Wang, "A noc traffic suite based on real applications," in *IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2011, 4-6 July 2011, Chennai, India*. IEEE Computer Society, 2011, pp. 66–71. [Online]. Available: https://doi.org/10.1109/ISVLSI.2011.49