

Area Optimization with Non-linear Models in Core Mapping for System-on-Chips

Jan Moritz Joseph¹, Dominik Ermel¹, Tobias Drewes¹, Lennart Bamberg², Alberto García-Ortiz², and Thilo Pionteck¹

¹Otto-von-Guericke-Universität Magdeburg, Institut für Informations- und Kommunikationstechnik, 39106 Magdeburg, Germany, Email: {jan.joseph, dominik.ermel, tobias.drewes, thilo.pionteck}@ovgu.de

²University of Bremen, Institute of Electrodynamics and Microelectronics, 28359 Bremen, Germany, Email: {agarcia, bamberg}@item.uni-bremen.de

March 12, 2021

Abstract

Linear models are regularly used for mapping cores to tiles in a chip. System-on-Chip (SoC) design requires integration of functional units with varying sizes, but conventional models only account for identical-sized cores. Linear models cannot calculate the varying areas of cores in SoCs directly and must rely on approximations. We propose using non-linear models: Semi-definite programming (SDP) allows easy model definitions and achieves approximately 20% reduced area and up to 80% reduced white space. As computational time is similar to linear models, they can be applied, practically.

Keywords: Design Models, Nonlinear Optimization, CAD

1 Introduction

A common design problem for System-on-Chips (SoCs) using mesh-based Network-on-Chips (NoCs) is core mapping. This takes a core graph and a network graph as input. In the core graph, nodes represent cores, edges represent communication and edge weights represent required bandwidths. In the network graph, nodes represent tiles, each of which is a NoC router with reserved space for a core, and edges represent links between tiles. The reserved area per core is usually identical. The output of core mapping is an assignment of cores to tiles. The objective function of core mapping minimizes communication costs,

e.g. required bandwidth times hop distance. This is solved using linear models, for instance mixed-integer linear programming (MILP), or heuristics such as simulated annealing (SA), e.g. [1].

The aforementioned approach does not account for cores with varying sizes due to different functionality, which is a typical characteristic of SoCs. Efficient core mapping for SoCs must account for this heterogeneity. An example for this inefficiency is depicted in Fig. 1, in which cores of different sizes (orange) allocate less area than reserved (light gray). Naturally, conventional linear models for mapping will be limited if the objective function also takes area into consideration, since area calculations are non-linear, intrinsically.

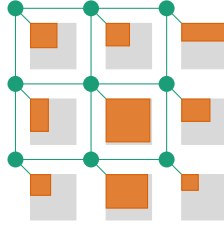


Figure 1: Mapping of cores (orange) of varying size to a SoC with equal-sized tiles. Whitespace is gray.

In this publication, we propose to use non-linear models for area minimization during mapping, since these yield better area with short run time. We demonstrate this by comparison of a linear and a non-linear model for area optimization during core mapping: The proposed non-linear model yields better area, since an error-inducing linearization of area is not necessary. Further, they are cleaner than linear models and their definition is less complicated than usually anticipated. Plus, the additional computational effort for optimization is very small. Due to this, non-linear models are highly relevant in practical application.

This publication is structured as follows: First, we contribute two models for minimization of area for mapping in Sec. 2; one linear and one non-linear model. Next, we compare the results of linear and non-linear models in Sec. 3.1. Then, the results of a mapping algorithm using SA, which considers area using non-linear models, are shown in Sec. 3.2 and compared to related work. Results are discussed in Sec. 4. Finally, we conclude in Sec. 5.

2 Area for mapping on cores

We introduce two basic models, which allow optimizing the area of a SoC during core mapping. The first model is linear and therefore approximates area. The second is non-linear and therefore more area-efficient. These two models are the basis, on which we compare linear and non-linear models.

The area optimization problem is formulated as follows: Assume a given order of lk or fewer components in l rows and k columns. This represents a

given mapping of cores to tiles. Each component has the size $F_{i,j}$, for certain $i \in [l] := \{1, \dots, l\}$ and $j \in [k] := \{1, \dots, k\}$. $F_{i,j} = 0$ if there is no component at row i , column j for all pairs $(i, j) \in [l] \times [k]$. The height of rows is denoted by $r_i \in \mathbb{R}$ for all $i \in [l]$. The width of columns is denoted by $c_j \in \mathbb{R}$ for all $j \in [k]$. It holds that the reserved area in a tile is bigger than the size of the assigned core:

$$r_i c_j \geq F_{i,j} \quad \text{for all } i \in [l], j \in [k]. \quad (1)$$

The linearized objective function throughout this paper is to minimize the side length of a square that encloses all tiles.

$$\max \left(\sum_{i \in [l]} r_i, \sum_{j \in [k]} c_j \right) \rightarrow \min \quad (2)$$

2.1 Linear Model

The area of a rectangle $F_{i,j}$ with edge length r_i and c_j cannot be calculated through means of a linear model. Therefore, the area is linearized. A natural approach is given by *Lacksonen et al.* [2] for a factory layout problem, which can be applied here as well. The approach is shown in Fig. 2. Linearization is possible, since rectangles are within an aspect ratio of $\eta \in (0, 1)$, i. e. $r_i \geq c_j \eta$ and $r_i \leq c_j \eta^{-1}$. This is shown in Fig. 2a. The area $r_i c_j$ of a tile i, j must be larger than its core with size $F_{i,j}$, i. e. $F_{i,j} \leq r_i c_j$. The hyperbola for equal area is the lower left bound for the solution space of the optimization. Further, the solution space is limited by any given maximum edge length of the tile, i. e. $c_j \leq y_{max}$ and $r_i \leq x_{max}$. The solution space is further reduced in size by the line equations for the aspect ratio η . Since the hyperbola is non-linear, it is approximated by a line equation given by the intersections between the lines for the aspect ratios and the maximum edge length. The resulting linearization error is plotted in green in Fig. 2a. It is possible to reduce this error by including more equally-spaced knots as shown in Fig. 2b. Each linear equation connecting two adjacent knots intersected with the iso-area-hyperbola ($r_i c_j = F_{i,j}$) is called a 1-spline. Please note that more 1-splines reduce the error but increase the model complexity, since integer inequalities are required to determine the current spline. There are at least three additional integer inequalities per supporting point. Naturally, this has a large performance impact.

For a given mapping, the optimization is subject to

$$r_i \geq \eta c_j \quad \forall i \in [l], \forall j \in [k] \quad (3)$$

$$c_j \geq \eta r_i \quad \forall i \in [l], \forall j \in [k] \quad (4)$$

$$r_i + c_j \geq \sqrt{F_{i,j} \eta} + \sqrt{F_{i,j} / \eta} \quad \forall i \in [l], \forall j \in [k] \quad (5)$$

with the minimum tile aspect ratio $\eta \in (0, 1)$. The linearization of Eq. 1 is now Eq. 5, following the approach presented using only a single linear approximation (cf. Fig. 2a).

2.2 Non-Linear Model

Since linear models introduce a significant error, we use a semi-definite model to optimize the problem without this linearization error, because the red hyperbola in Fig. 2 is modeled. We set kl variables $X_{k(i-1)+j}$ such that

$$X_{k(i-1)+j} = \begin{bmatrix} r_i & \sqrt{F_{i,j}} \\ \sqrt{F_{i,j}} & c_j \end{bmatrix} \succeq 0, \quad \forall i \in [l], \quad \forall j \in [k] \quad (6)$$

These matrices are premised to be positive semidefinite (i.e. " $\succeq 0$ "); thus each principal minor is greater or equal to 0:

$$\det(X_{k(i-1)+j}) \geq 0 \quad (7)$$

$$\Leftrightarrow r_i c_j - F_{i,j} \geq 0 \quad (8)$$

$$\Leftrightarrow r_i c_j \geq F_{i,j}, \quad \forall i \in [l], \quad \forall j \in [k] \quad (9)$$

We formulate a semi-definite programming (SDP) problem. The objective function minimizes a variable $x \geq \max\{\sum r_i, \sum c_i\}$ subject to:

We assign the corresponding area values to each matrix using the Frobenius inner product:

$$\begin{aligned} 2\sqrt{F_{ij}} &\leq \left\langle \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, X_{k(i-1)+j} \right\rangle \\ &\leq 2\sqrt{F_{ij}}, \quad \forall i \in [l], \forall j \in [k] \end{aligned} \quad (10)$$

For each $i \in [l]$, the upper left entry of the matrices $X_{k(i-1)+j}$ has the same value for all $j \in [k]$ (this models r_i):

$$\begin{aligned} 0 &\leq \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, X_{k(i-1)+1} \right\rangle + \\ &\quad \left\langle \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}, X_{k(i-1)+j} \right\rangle \leq 0 \end{aligned} \quad (11)$$

For each $j \in [k]$, the lower right entry of the matrices $X_{k(i-1)+j}$ has the same value for all $i \in [l]$ (this models c_j):

$$\begin{aligned} 0 &\leq \left\langle \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, X_j \right\rangle + \\ &\quad \left\langle \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}, X_{k(i-1)+j} \right\rangle \leq 0 \end{aligned} \quad (12)$$

We model the maximum variable x for the objective function (this models $x \geq \sum r_i$ and $x \geq \sum c_i$):

$$0 \leq x + \sum_{i=1}^l \left\langle \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}, X_{k(i-1)+1} \right\rangle \quad (13)$$

$$0 \leq x + \sum_{j=1}^k \left\langle \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}, X_j \right\rangle \quad (14)$$

Again, areas of tiles are constrained by an aspect ratio η . Note, that this aspect ratio is not violated by the relation between r_i and c_j . Rather, a component can find a rectangle inside the bounding box given by $r_i c_j$. This rectangle has the size of the core. The aspect ratio of its edges is greater than η . We formulate for all $i \in [l]$ and for all $j \in [k]$:

$$\sqrt{\eta F_{i,j}} \leq \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, X_{k(i-1)+1} \right\rangle \quad (15)$$

$$\sqrt{\eta F_{i,j}} \leq \left\langle \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, X_j \right\rangle \quad (16)$$

3 Results

We first compare the provided linear and non-linear model using theoretical benchmarks to quantify advantages for area and analyze additional effort in computational time in Sec. 3.1. Then, we apply the non-linear model to mapping of cores with different sizes in a SoC and compare the results to the related work in Sec. 3.2.

3.1 Linear vs. non-linear model

We compare the results of linear and non-linear models using the proposed LP and SDP. The models are implemented in MATLAB R2018a. LPs are optimized using IBM CPLEX 12.8.0 [3]. SDPs are optimized using Mosek 8.1 [4].

We generate three input benchmarks. The cores are equal-sized to provide a fair comparison against conventional approaches. The benchmarks are:

1. A 3D SoC with 2 layers and 5 tiles, of which three tiles are in layer 1 and two tiles are in layer 2.
2. A 3D SoC with 4 layers and 10 tiles per layer connected by a 2×5 mesh NoC.
3. A 3D SoC with 4 layers and 20 tiles per layer connected by a 4×5 mesh NoC.

Cores are 10 A large. Routers with 5 ports require 1 A . The router area is linearly proportional to port count depending on the position of the router in the network. TSV arrays, which vertically connect routers, are 2 A large. The aspect ratio is limited by $\eta = 0.1$. We run the optimization 50 times to average run time on an Intel Core i7-6700 (4 physical cores at 3.4 GHz with hyper threading) using Windows 10.

Both models require approximately the same amount of memory in MATLAB. The area and runtime results are reported in Table 1. In benchmark 1), the summed chip area is 68.7 A from the LP and 59.8 A from the SDP. In benchmark 2), the summed chip area is 832 A from the LP and 695 A from the

Table 1: Area, runtime, inequality count and variable count comparison between linear and non-linear model to optimize a homogeneous 3D SoC with 5, 40 and 80 homogeneous cores including TSV areas (runtime average of 50 reruns).

LAYER	AREA [A] AND DIFFERENCE								
	5 PEs			40 PEs			80 PEs		
	LP	SDP	Δ	LP	SDP	Δ	LP	SDP	Δ
1	43.0	36.8	-14.4%	211	178	-15.6%	364	301	-17.4%
2	25.7	23.0	-10.5%	222	180	-18.9%	379	313	-17.4%
3	—	—	—	214	183	-14.5%	378	313	-17.2%
4	—	—	—	185	154	-16.8%	316	261	-17.4%
AVERAGE AREA REDUCTION			-12.5%	-16.5%			-17.3%		
AVERAGE RUNTIME [s]									
	0.4	2.9	+625%	3.9	7.5	+92.3%	12.2	16.0	+31.1%
INEQUALITY COUNT									
	16	31	+94%	88	436	+395%	168	1644	+879%
VARIABLE COUNT AND DIFFERENCE [%]									
	9	21	+133%	32	112	+250%	40	200	+400%

SDP. In benchmark 3), the summed chip area is 1272 A from the LP and 1188 A from the SDP. Since in the lowest layer there is no TSV area required (there are no keep-out-zones using via-middle-process-flow), this layer is smaller. The difference in run time between LP and SDP is smaller for larger input sets.

The linear model requires $2kl + 2$ inequalities and $k + l + 1$ variables. The non-linear model requires $(kl)^2 + k + l + 2$ inequalities and $kl + 1$ variables.

3.2 Mapping with linear vs. non-linear models

Quadratic-shaped cores of a different size are mapped in a 2D mesh NoC in [1] using a linear model and a heuristic for larger input sets. Their objective function does not target low area but minimizes transmission energy. We compare the results from our non-linear model with results from linear models from [1] using the three benchmarks provided (see Table 2). The data streams for the benchmarks are taken from [5] and the cores' area from [1].

We implement a simple SA for mapping: The neighbor function randomly changes the position of one core in the mapping. If the position is already taken by another core, both will be switched. We extend the objective function from [1] to include chip area. The initial solution for our SA places cores decreasingly ordered by size, such that white space is minimized and communication is not accounted for.

The results for area and network performance are given in Table 2. The sum of all tiles' sizes gives the total area (including whitespace). The network performance is measured by accumulated link load (measuring delay) and maximum link load (measuring throughput). Four data sets are given per benchmark: First, the baseline from [1] for a 2D mesh NoC. Second, the first configuration is optimized using the proposed non-linear model without changing the map-

Table 2: Area and network performance comparison of mapping to a 2D-mesh NoC with [1]. The SA is executed with 20 reruns, an initial temperature of 30, cooling of 0.98 and 15,000 iterations. The aspect ratio is limited by $\eta = 0.1$.

			AREA [A]			COMMUNICATION [Bits*Hops]			BANDWIDTH [Bits]		
			MEAN	STD	RATIO	MEAN	STD	RATIO	MEAN	STD	RATIO
H256 DEC	mp3 DEC	BASLINE	11301	—	—	19858	—	—	4060	—	—
		[1]	11178	—	-9.94%	19858	—	0.0%	4060	—	0.0%
		WITH SDF	7902	—	-30.1%	33707	—	+69.7%	7994	—	+96.9%
		INITIAL	8244	505	-27.1%	21280	624	+7.16%	4452	674	+9.66%
H263 ENC	mp3 DEC	BASLINE	12535	—	—	255324	—	—	84884	—	—
		[1]	10178	—	-18.8%	255324	—	0.0%	84884	—	0.0%
		WITH SDF	6993	—	-44.2%	525537	—	+106%	85244	—	+0.42%
		INITIAL	10474	2148	-16.4%	250187	14763	-2.0%	73161	17497	-13.8%
mp3 ENC	mp3 DEC	BASLINE	8568	—	—	17546	—	—	4085	—	—
		[1]	8091	—	-5.57%	17546	—	0.0%	4085	—	0.0%
		WITH SDF	7281	—	-15.0%	39171	—	+123.3%	6560	—	+60.1%
		INITIAL	8516	796	-0.61%	17572	487	+0.15%	4974	902	+21.8%

ping. This quantifies the potential of non-linearity. Third, the initial solution for the SA is given, which is area-efficient and communication-inefficient. Fourth, our SA is executed 20 times with 15,000 iterations, an initial temperature of 30 and a cooling of 0.98. The aspect ratio is limited by $\eta = 0.1$. The results of all runs are averaged and the standard deviation is calculated. We balance the weights in the cost function and prioritize neither area nor communication. A single run of the SA terminates after approximately 17 minutes on a Windows 10 workstation using an Intel i7-7740X processor (4 physical cores at 4.3 GHz with hyper threading).

4 Discussion

4.0.1 Area

The area of non-linear models is better, as expected. The theoretical benchmarks, see Table 1, show their clear advantage in terms of area minimization. Area is reduced between 10.5% and 18.9%. The results for the real-world based benchmarks further support this because the area of the mapping is reduced by 5.57% to 18.8% in comparison to the solution from [1], see Table 2. If declined communication efficiency is acceptable, the area can further be reduced. For instance, the H256 dec mp3 dec can be mapped with 27.1% area reductions at the expense of 7.16% worse communication costs and 9.66% worse bandwidth over baseline. The other two benchmarks show similar results. Summing up, non-linear models enable area reductions of up to 27% in our benchmarks.

4.0.2 Model definition

The proposed model definition supports, that non-linear models are cleaner than linear models: First, error-prone linearization is not required. Second, the SDP directly models the area multiplication, which increases readability and allows for easier understanding of the problem. Therefore, the model is less complicated. Further, the effort to define a non-linear model is small. The whole model is given by 10 inequalities.

4.0.3 Optimization run time

Higher expected run time is a common reason to reject non-linear models. Our results do not support this. Of course, the run time of the non-linear model is slightly worse than the linear model. But the SDP only requires 16 seconds to find an optimal solution even for the large example with 80 cores. The variable count grows fast for the non-linear model (SDP: $\Omega((kl)^2)$ vs. LP: $\Omega(k + l)$) but this does not have a negative impact on performance. The number of inequalities grows similar for both at the rate of $\Omega(nm)$ and the optimizer for the SDP handles this better. The run time of the SA in the real-world based benchmarks of 17 minutes is reasonable for daily usage. This demonstrates that non-linear models are applicable in practice.

5 Conclusion

In this paper we show that using non-linear models for area optimization in core mapping yields better area with better-structured model definitions than conventional linear models. We contribute both a linear model (LP) and a non-linear model (SDP) to reduce the amount of white space in reserved areas for cores. Comparison of the two models shows that area can be reduced by up to 27%. Since the non-linear model does not require area approximation, the model definition is cleaner. A small set of inequalities directly describes the

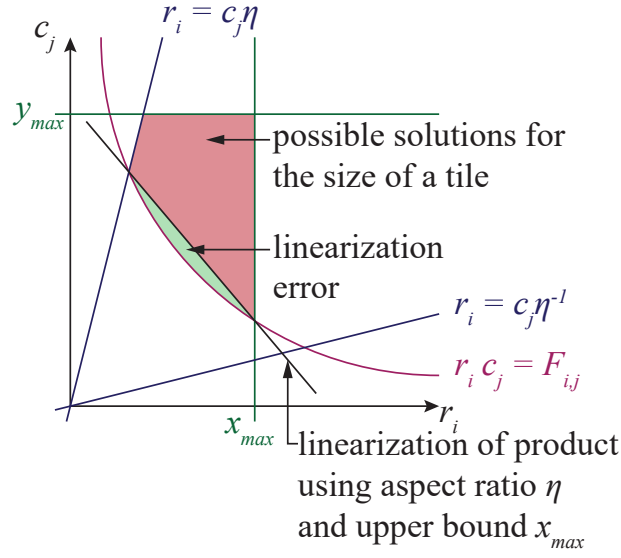
problem completely. For a large benchmark example with 80 cores, an optimal solution is found within 16 seconds.

Acknowledgments

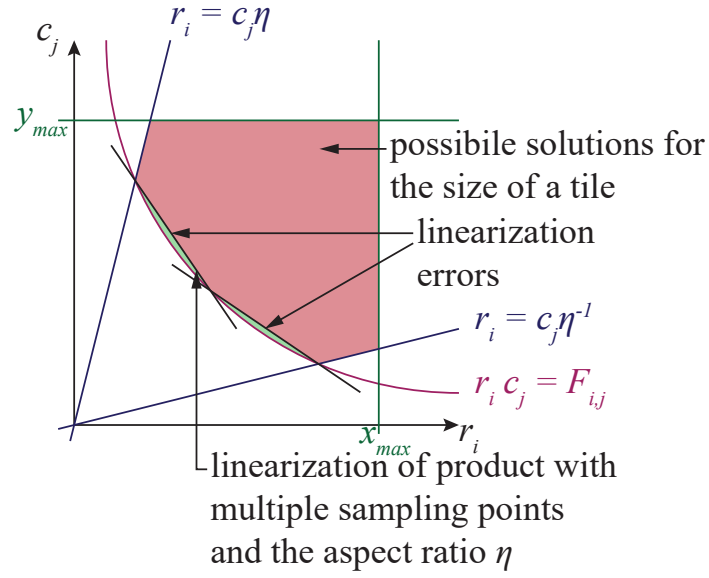
This work is funded by the German Research Foundation (DFG) project PI 447/8-1.

References

- [1] K. Srinivasan, K. S. Chatha, and G. Konjevod, “Linear-programming-based techniques for synthesis of network-on-chip architectures,” *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 4, pp. 407–420, 2006.
- [2] T. A. Lacksonen, “Static and Dynamic Layout Problems with Varying Areas,” *Journal of the Operational Research Society*, vol. 45, no. 1, pp. 59–69, 1994.
- [3] IBM, “Cplex 12.8 User’s Manual,” 2017.
- [4] Mosek ApS, “Mosek,” 2018.
- [5] P. K. Sahu and S. Chattopadhyay, “A survey on application mapping strategies for Network-on-Chip design,” *Journal of systems architecture*, 2013.



(a) Simple approximation with single linear equation.



(b) Reduced error through multiple linear approximations.

Figure 2: Area linearization.