A Graph-Based Model of Micro-Transfer Printing for Cost-Optimized Heterogeneous 2.5D Systems

Robert Fischbach, Tilman Horst and Jens Lienig
Dresden University of Technology, Institute of Electromechanical and Electronic Design
Dresden, Germany, www.ifte.de
Email: [robert.fischbach][tilman.horst][jens.lienig]@tu-dresden.de

***Abstract*—Micro-transfer printing (µTP) is a promising assembly technology that enables heterogeneous integration of dies originating from different wafers. It combines the advantages of pick-and-place in terms of flexibility with the advantages of wafer-level processing in terms of high throughput. µTP applies an elastomer stamp to transfer multiple dies from source to target wafers in parallel. Increasing the stamp size allows for the transfer of more dies at once and reciprocally shortens the manufacturing time, enabling extensive cost reductions. On the other hand, larger stamps result in a lower wafer utilization, thereby causing increases in costs. Finding the cost-optimal stamp layout is one of the key tasks when designing heterogeneous systems for µTP. There is no trivial solution to calculate the wafer utilization needed to evaluate the quality of a stamp layout. Based on a graph problem known as maximum independent set, we propose a model to determine the wafer utilization subject to the stamp and wafer layout. We demonstrate the application of our model within an economic cost function to optimize a µTP design with regard to manufacturing costs.**

I. Introduction

Heterogeneous systems consist of multiple components closely integrated at wafer-level and provide new opportunities for the next generation of microelectronics. Such systems are not restricted to a specific technology, such as CMOS. Instead, different highly specialized semiconductor processes can be combined, each utilizing their unique properties and advantages. Areas of application are automotive and biomedical sensors, opto-electronics, µLEDs and more. Apart of computational power provided by CMOS, such applications require sensing, fast switching or illumination capabilities, which are often beyond the scope of CMOS. Therefore, heterogeneous integration of different processes is expedient and has proven viable in laboratory and pilot line conditions. However, market shares of heterogeneous systems are still low due to high manufacturing costs. The integration of heterogeneous systems is challenging: different components of such systems are designed and manufactured independently, but eventually have to work as a single design- and cost-optimized unit. This urgently demands appropriate chip/package co-design flows [1]–[4].

In this paper, we present a promising assembly technology, named micro-transfer printing (µTP) (Fig. 1), which considerably lowers manufacturing cost for heterogeneous systems compared to alternative approaches like pick-and-place. For designers, however, it is hardly possible to make use of µTP's cost savings capability without appropriate design support tools.

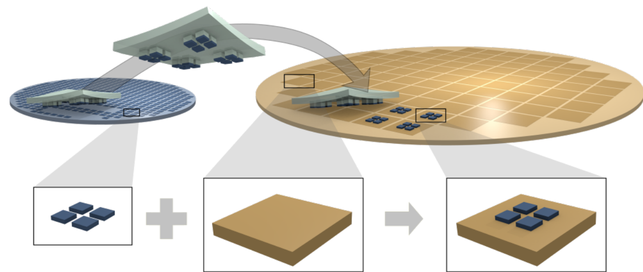

Figure 1. The major steps of micro-transfer printing (µTP): picking up dies from a source wafer (left) that carries different kinds of components to be integrated, transfer via an elastomer stamp (top) and release on a target wafer (right), which contains the dies that serve as final carrier.

To solve this problem, we propose a cost model to calculate the unit cost for a heterogeneous system manufactured by µTP as well as a graph-based model to determine and optimize wafer utilization. This enables the design of cost-optimized heterogeneous systems manufactured by µTP. The presented approach is intended as a first step towards models that support manufacturing-cost-based evaluations of design and process parameters and their optimization.

This paper is an extended version of [5] and is organized as follows. Section II provides a short introduction to µTP as a foundation for the models developed in the subsequent chapters. The modeling of µTP with regard to the wafer utilization is described in Section III; the cost model is presented in Section IV. Section V presents the implementation of these models and their application to a cost optimization problem in heterogeneous system design.

II. Micro-Transfer Printing

With regard to manufacturing of heterogeneous systems, µTP is a promising assembly technology as it combines the advantages of pick and place in terms of flexibility with the advantages of wafer-level processing in terms of high throughput [6], [7]. Figures 1 and 2 illustrate the µTP process with its three main constituents.

The *source wafer* can carry different components (such as passive or active devices), referred to as *source dies*, to be integrated into a heterogeneous system. In order to release the dies from the source wafer, wet chemical undercut etching is performed prior to the actual transfer process [8], [9].

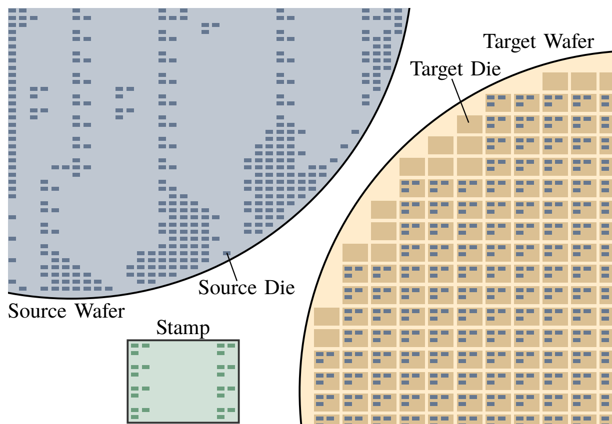

Figure 2. During µTP, several source and target dies remain unprocessed. The degree of such disuse is expressed as wafer utilization and strongly depends on the stamp layout.

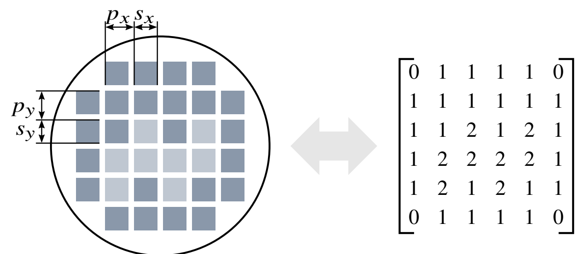

Figure 3. A wafer layout is represented as a matrix discretized by the die pitch (p_x and p_y). The pitch as well as the die size (s_x and s_y) coincide with the elements on the stamp.

The *target wafer* comprises the *target dies* on which the source dies shall be printed. Thus, the target die serves as carrier and is therefore usually the largest component of the resulting die stack. Typically, an adhesion layer is deposited on the target wafer before printing. The target wafer does not require a special treatment with regard to separation as it will undergo conventional wafer dicing.

The µTP process utilizes a micro-structured elastomer *stamp*, capable to pick and transfer a large number of source dies (>10,000) to the target wafer. The printing process is based on van der Waals forces between the source dies and the stamp. The pickup and release process can be controlled due to a stamping-speed-depending adhesion between the source dies and the stamp. During a fast movement of the stamp, its adhesion is larger than the bonding with the wafer, and vice versa. Depending on the size and the layout of the stamp, not all source and/or target dies are accessible. The degree of which a wafer can be accessed by the stamp is called *wafer utilization*.

The µTP process ends with the placement of the source dies, followed by processing steps on (target) wafer-level, such as the creation of electrical interconnects via a redistribution layer (RDL).

The main benefits of µTP are as follows: substrate-based as well as substrate-less stacking of heterogeneous components on package-level; source and target dies as well as the wafers can be of arbitrary sizes; highly parallelized transfer process, with the option of originating from multiple source wafers; subsequent processing on wafer-level.

µTP introduces a manufacturing-relevant stamp layout, which is strongly interrelated with the chip and package layouts. Novel layout dependencies between source, target and stamp have to be taken into account – focusing on their strong impact on manufacturing costs. These interdependencies must be considered in a chip/package co-design flow.

In µTP, the manufacturing costs depend on the utilization of the source and target wafers as well as on the manufacturing throughput. The goal is to find design parameters that have the lowest manufacturing costs and that meets all relevant electrical constraints. The design example in Figure 2 is based on a set of parameters, such as the layouts of the source, target and stamp. Irrespective of whether the design parameters are determined with the aid of an optimization procedure or are specified manually, the evaluation of each solution must be possible in order to make credible design decisions. Usually, such an evaluation is implemented by a cost function combining one or more cost criteria, such as manufacturing costs. Additionally, µTP-designs require the knowledge of the utilization of the source and target wafers (Sect. III), as it is an essential part of the cost model (Sect. IV).

III. Wafer Utilization

The wafer utilization is the ratio between the picked-up dies and the total number of dies on a wafer. The number of unpicked dies depends on the specific positions of the sequentially applied stampings. An appropriate heuristic to estimate the wafer utilization is needed, as the optimization of the stamp positions on a wafer to maximize its utilization is a NP-hard problem (see Sect. III-B).

A. Problem Formulation

As motivated above, our goal is to provide a heuristic to determine wafer utilization of a given stamp and wafer combination. Essentially, we need to find an optimal set of stamping positions in such a way that those positions are valid and the number of addressed dies on a wafer is maximized. The corresponding algorithm is described in Section III-B and works with following abstraction.

As the stamp and the (source) wafer have identical grid and element sizes, the wafer utilization can be determined independently of these parameters; only the relative layouts of the elements matter. Thus, the wafer and stamp layouts, which are required as input to the algorithm, can be reduced to discretized matrices as illustrated in Figure 3.

Basically, each element in the layout corresponds to an entry in a matrix $M^{r \times c} = (m_{i,j})$, where the rows and columns represent the layout grid. An entry $m_{i,j}$ corresponds to a layout position $(x, y) = ((j-1) \cdot p_x, (i-1) \cdot p_y)$. We set $m_{i,j} = 1$ if an element exists at that position (e.g., a die on the source wafer), $m_{i,j} = 0$ if there is no element at that position, and $m_{i,j} = 2$ if the element is "picked" (relevant for wafers only).

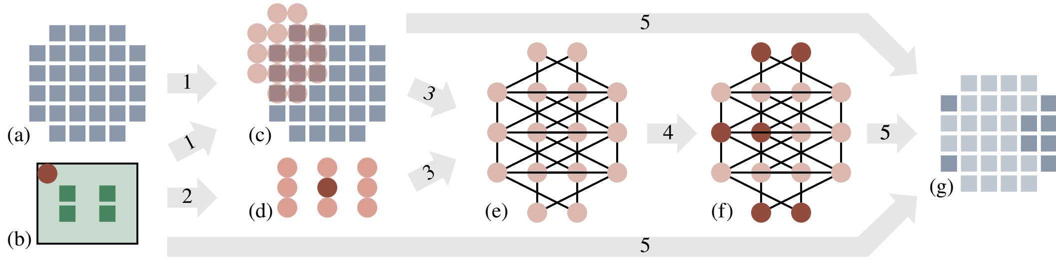

Figure 4. Steps of the wafer utilization algorithm (detailed description in Section III-B): (1) The matrix representations of the wafer (a) and stamp (b) yield the valid stamp positions (c). (2) The stamp layout induces layout dependencies (d). (3) The valid stamp positions and these dependencies are represented as a dependency graph (e). (4) Addressing this graph as a maximum independent set problem, results in the maximum number of stampings (f). (5) The resulting set of nodes is interpreted as stamped positions, determining the possible wafer utilization (g).

The layout coordinates originate at the top left with x increasing in positive horizontal direction and y increasing in negative vertical direction. The utilization can easily be determined by counting the picked elements in $M^{r \times c}$.

Throughout the paper, the following assumptions and simplifications are made: the target die pitch is a multiple of the source die pitch; each stamp needs to be fully populated; usage of a single stamp only (i.e., no repair steps, no stamp combinations); wafer layouts do not contain any auxiliary structures (e.g., alignment markers, test structures); no consideration of known good die or yield models.

B. Algorithm Description

Figure 4 illustrates the wafer utilization algorithm. Input to the algorithm is the layout of the wafer (Fig. 4a) and the stamp (Fig. 4b). Basically, the provided layout data is reduced to a Maximum Independent Set (MIS) problem and (heuristically) solved by KaMIS, a third party MIS solver [10]. Finally, our algorithm outputs the utilization of the wafer (Fig. 4g) and is divided into the following five steps (enumeration as in Fig. 4):

(1) Determination of Valid Stamp Positions: The first step is to identify all valid stamp positions (Fig. 4c). In our current setup we assume only fully populated stamps as valid[1]. Valid positions are derived from a “simulated” application of the stamp on the wafer. Note that the resulting stamp positions may have negative indices with regard to the wafer matrix.

First, we create two sets W (Eq. 1) and S (Eq. 2) containing all wafer and stamp elements, respectively. Based on W and S, the valid stamp indices V (Eq. 3) are obtained.

$$W \leftarrow \{(i, j) \mid m_{i,j} = 1 \wedge m_{i,j} \in M_{\text{Wafer}}\} \quad (1)$$

$$S \leftarrow \{(i, j) \mid m_{i,j} = 1 \wedge m_{i,j} \in M_{\text{Stamp}}\} \quad (2)$$

$$V \leftarrow \{(i, j) \mid i, j \in \mathbb{Z}, \forall (i_s, j_s) \in S \ \exists (i + i_s, j + j_s) \in W\} \quad (3)$$

(2) Identify Stamp Implications: In order to find out how a stamping on one particular position $v_m \in V$ invalidates other positions, we analyze S and obtain the first order dependencies. A stamp on a position v_m would pick some elements from the wafer. In consequence, a stamp position v_n which also requires one of these already-“picked” elements is related to v_m. We store these directed dependencies within another set D (Eq. 4), which contains the affected positional offsets (Fig. 4d). Note the point reflection of the resulting (directed) dependency offsets. Since we will target undirected relations between each stamp position in the next step, this symmetry can be used to reduce the number of offsets by half.

$$D \leftarrow \{(i_o, j_o) \mid i_o, j_o \in \mathbb{Z}^*, \exists (i_s, j_s) \in S \ \exists (i_t, j_t) \in S : (i_s + i_o, j_s + j_o) = (i_t, i_t)\} \quad (4)$$

(3) Building the Stamp Dependency Graph: Based on D and V, we derive a graph G (Eq. 5) which represents the dependencies between different stamp positions (Fig. 4e). Each node in this graph maps to one valid stamping position. If a stamp dependency between two positions exists, an edge is inserted between the two corresponding nodes (Eq. 6).

$$G = (V(G), E(G)), V(G) \leftarrow V \quad (5)$$

$$E(G) \leftarrow \{e \mid \Psi_G(e) = (u, v) = (v, u), u, v \in V(G), \exists d \in D : u + d = v\} \quad (6)$$

(4) Solving the Maximum Independent Set Problem: At this point, the wafer utilization problem is reduced to a form where it can be addressed as a known mathematical problem. In order to get the maximum number of stampings on the wafer, we need to identify the maximum number of independently selectable nodes in G (i.e., all selected nodes must not share a single edge, Fig. 4f). This NP-hard problem, known as a maximum independent set (MIS) problem, is solved by applying KaMIS, a solver for the MIS problem [10]. KaMIS returns the desired maximum independent set V_{MIS} (Eq. 8).

$$V_{\text{IS}} \leftarrow \{V' \subseteq V \mid \nexists u, v \in V', e \in E(G), \Psi_G(e) = (u, v)\} \quad (7)$$

$$V_{\text{MIS}} \leftarrow \{V' \in V_{\text{IS}} \mid \nexists V'' \in V_{\text{IS}} : |V''| > |V'|\} \quad (8)$$

(5) Apply Stampings on Wafer: With V_{MIS} available, it is straightforward to apply the corresponding stampings on the wafer (Fig. 4g). For each element position on the stamp, the respective stamp position offset is applied.

$$W_S \leftarrow \{(i, j) \mid \forall (i_p, j_p) \in V_{\text{MIS}} \ \forall (i_s, j_s) \in S, (i, j) = (i_p + i_s, j_p + j_s)\} \quad (9)$$

[1]It is also possible to consider partially populated stamps as valid. However, this would require a transition towards a maximum weighted independent set problem which is not within the scope of this paper.

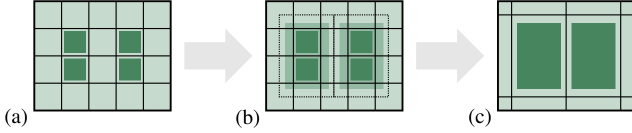

Figure 5. Algorithmic adaptation of the stamp layout. The matrix representation of the original stamp layout (a) is directly used to determine the source wafer utilization. The necessary (algorithmic) adaptation to the target grid (c) is done by matching it with the target die layout (b).

C. Adaptation to Target Wafer

The presented algorithm can be applied to obtain the source wafer utilization as each element on the stamp directly corresponds to an element on the source wafer. In contrast, the target wafer utilization can not be calculated directly with the presented algorithm; instead, a slight modification is required. Specifically, we need to create a virtual “target wafer stamp” on which the element grid corresponds to the target wafer grid.

Figure 5 illustrates the conversion. A given (source) stamp is partitioned corresponding to the source wafer grid (see Fig. 5a). In consequence of the µTP process, the stamp also shows an implicit second order pattern (i.e., the repeating target die layout). This can be seen in Fig. 5b where two source dies are placed on each target die. These sub-layouts result in a new (target die) grid which yields the required “target wafer stamp” (see Fig. 5c).

If this derived stamp is used in combination with the target wafer layout as input, the previously presented algorithm determines the target wafer utilization.

IV. Cost Model

The following cost model focuses on manufacturing costs and is used to motivate the modeling of assembly-related processes (in our case µTP). This model is simplified for better comprehensibility and therefore ignores technological details, such as the required changeover times (for stamp or wafer exchange in the printing tool). In the following, n_x stands for “number of x”, c_x for “costs per x”, u_x for “utilization of x” and t_x for “time per x”, respectively.

To calculate the final cost (per piece) of a heterogeneous system manufactured using µTP, our cost function (Eq. 10) incorporates the unit costs of the target die $c_{\text{TargetDie}}$, the unit costs of the source die $c_{\text{SourceDie}}$ and the costs resulting from assembling the system $c_{\mu\text{TP}}$.

$$cost = c_{\text{TargetDie}} + c_{\text{SourceDie}} \cdot n_{\text{SourcesDiesPerTarget}} + c_{\mu\text{TP}} \quad (10)$$

The cost per target die (Eq. 11) is calculated by dividing the costs of an target wafer $c_{\text{TargetWafer}}$ by the number of usable dies per wafer, which is the product of the total number of dies per wafer $n_{\text{DiesPerTargetWafer}}$ and the corresponding target wafer utilization $u_{\text{TargetWafer}}$. The cost per source die $c_{\text{SourceDie}}$ is calculated accordingly (Eq. 12).

$$c_{\text{TargetDie}} = \frac{c_{\text{TargetWafer}}}{n_{\text{DiesPerTargetWafer}} \cdot u_{\text{TargetWafer}}} \quad (11)$$

$$c_{\text{SourceDie}} = \frac{c_{\text{SourceWafer}}}{n_{\text{DiesPerSourceWafer}} \cdot u_{\text{SourceWafer}}} \quad (12)$$

Table I
Parameters of the cost model.

Parameter	Value
$n_{\text{SourceDiesPerTarget}}$	4
$c_{\text{TargetWafer}}$	\$ 15,000
$n_{\text{DiesPerTargetWafer}}$	7845
$c_{\text{SourceWafer}}$	\$ 15,000
$n_{\text{DiesPerSourceWafer}}$	70681
$c_{\text{MachineHour}}$	\$ 1,000
t_{Stamping}	20 s

$u_{\text{TargetWafer}}$ and $u_{\text{SourceWafer}}$ are obtained by applying the algorithm described in Section III. Note that a credible cost estimation is not possible without knowing the utilization of the wafers during the manufacturing process.

The cost for the assembly of the system $c_{\mu\text{TP}}$ is calculated according to Eq. 13. The time-dependent costs for using the production line $c_{\text{MachineHour}}$ are divided by the number of target dies that can be covered during a single stamping $n_{\text{TargetDiesPerStamp}}$. This fraction is multiplied by the stamping duration per hour t_{Stamping}.

$$c_{\mu\text{TP}} = \frac{c_{\text{MachineHour}}}{n_{\text{TargetDiesPerStamp}}} \cdot t_{\text{Stamping}} \quad (13)$$

Most parameters that appear in the cost equations are given or can easily be derived. Table I lists the direct cost model parameters and their values as used in the cost optimization example in Section V-B. The number of target dies per stamp $n_{\text{TargetDiesPerStamp}}$ depends on the optimization parameter s (stamp size) and is calculated for each particular value.

The cost equations depict the trade-off between the numbers of dies per stamp (and thus, throughput) and the number of usable dies per wafers (i.e., the larger the stamp, the less stampings are required, but the lower the expected utilization will be). An increased wafer utilization can reduce manufacturing costs significantly. Hence, the optimization of the design parameters (e.g., stamp size) is required in order to apply µTP in an economically efficient manner.

V. Implementation

This section describes the implementation of our method and shows first results. The data preparation and transformation steps were implemented in Python (steps 1-3, 5 in Fig. 4), whereas the maximum independent set solver (KaMIS) was integrated via its command line interface (step 4 in Fig. 4). The soft- and hardware specifications of our development system are as follows: Python 3.7 with NumPy 1.6, KaMIS 1.0, running on an Intel(R) Core(TM) i7-7700K CPU with 16 GB system memory under Linux (kernel) 4.19.

A. Runtime Behaviour

In order to assess the runtime behaviour of our implementation, we generated test data for the (squared) stamp and wafer matrices depending on the parameters s (stamp size) and d (wafer diameter). s and d are normalized with regard to the die pitches on the corresponding wafer. We generate wafer

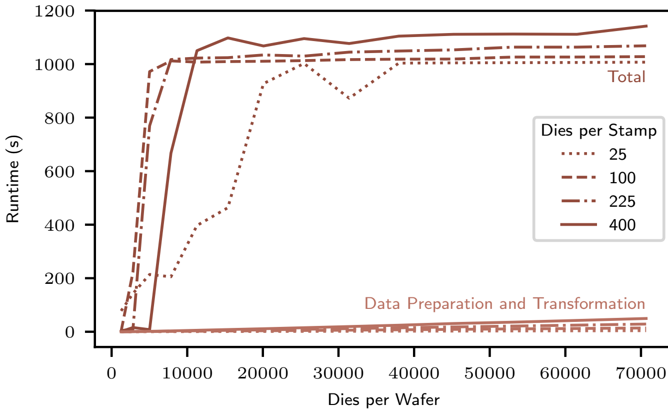

Figure 6. The runtime of the wafer utilization algorithm grows with increasing numbers of elements on wafer and stamp and characterized by the MIS-solver.

matrices $M_{\text{Wafer}} \in \mathbb{Z}^{d \times d}$ with its elements $m_{i,j}$ valued 1 in a wafer-like circular shape within the matrix and 0 otherwise.

$$m_{i,j} = \begin{cases} 1, & \text{if } (i-(d+1)/2)^2 + (j-(d+1)/2)^2 \leq d^2/4 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

The stamp matrix $M_{\text{Stamp}} \in \mathbb{Z}^{s \times s}$ with all elements set to "1" represents a fully populated square-shaped stamp of a certain size.

Based on this test data, we measured the runtime of our implementation over various wafer diameters and stamp sizes. The results in Figure 6 show the total runtime which is dominated by the KaMIS solver. Obviously, KaMIS quickly finds good solutions. However, as it is a probabilistic metaheuristic (evolutionary algorithm [11]), it continues its search for a better solution until it times out by default at around 1000 seconds. The runtimes of the data preparation and transformation within Python (steps 1-3, 5 in Fig. 4) are plotted separately. The runtime complexity of the Python code depends on the number of dies per wafer n as well as the number of dies on the stamp m. Generally, n is considerably larger than m, hence $O(n,m) = n \cdot m$ which corresponds with the experimental measurements. Note that n and m itself grow quadratically with increasing s and d, respectively.

On our development system, the (yet unoptimized and single-threaded) implementation of the presented approach is applicable to problem sizes up to 100,000 elements on the wafer and 1000 elements on the stamp. This scenario required approx. 10 GB RAM. In future we plan to improve our implementation (e.g., parallelization and memory-optimized data structures). This will enable challenging µTP designs with more than 300,000 dies on a single wafer and more than 10,000 elements per stamp.

Currently, our implementation is suitable for essential investigations of the µTP process. Furthermore, our approach can serve as a baseline algorithm for faster approximation-based models, applicable to interactive user interfaces and highly iterative optimization loops.

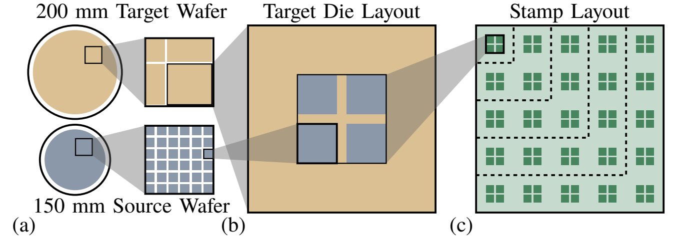

Figure 7. Layouts used in the cost optimization example. Wafer maps for source and target dies (a), target die layout with four printed source dies (b) and the corresponding stamp layout to transfer the dies from source to target (c). Several different stamp variants are indicated by the dashed lines.

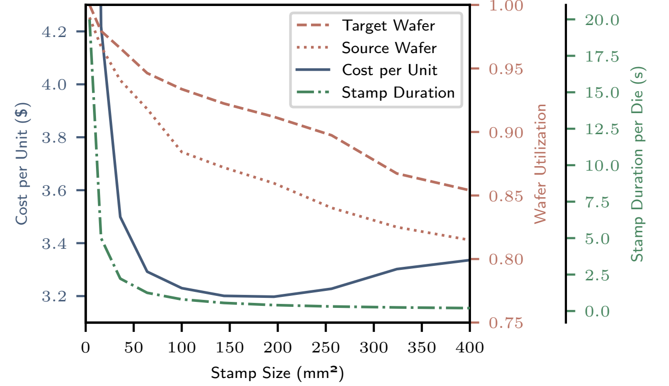

Figure 8. Cost per unit over stamp size. The lowest cost per unit is achieved by optimizing the trade-off between stamp duration and wafer utilization.

B. Cost Optimization

In this section we demonstrate the application of our models to a basic cost optimization of a stamp layout used to manufacture a µTP die stack. Figure 7 shows the example layout and Table II specifies its design parameters. The chosen design parameters reflect current manufacturing capabilities of the µTP technology and are oriented towards the pilot line demonstrators within the MICROPRINCE project [9], [12], [13]. Both, the dies on the 200 mm target wafer as well as those on the 150 mm source wafer are square-sized. The pitches of the respective dies are matched to satisfy the layout interdependency caused by the stamp, which requires compatibility to both wafers (source and target). In our case the target die pitch is four times the source die pitch. The stamp size s is the optimization parameter, constrained to a squared shape and a maximum

Table II

Design parameters used in the cost optimization example. Dies and stamp are constrained to squared shapes for simplification. Thus, single side length parameters are sufficient to describe these shapes.

Scope	Parameter	Value
Source Wafer	SW_{Diameter}	150 mm
	SW_{DieSize}	400 µm
	SW_{DiePitch}	500 µm
Target Wafer	TW_{Diameter}	200 mm
	TW_{DieSize}	1900 µm
	TW_{DieSize}	2000 µm
Stamp	S_{MaxSize}	20 mm

Algorithm 1: Optimization of the stamp size s.

Input: Parameters from Tables I and II, target die layout and wafer maps for source and target (Fig. 7)
Output: Optimized stamp size s_{opt} at minimal cost c_{opt}

```
d_SW ← ⌈SW_Diameter / SW_DiePitch⌉ ;            /* normalize */
d_TW ← ⌈TW_Diameter / TW_DiePitch⌉;
M_SW ∈ Z^(d_SW × d_SW) ← read(SourceWaferMap);
M_TW ∈ Z^(d_TW × d_TW) ← read(TargetWaferMap);
M_StampTile ← read(TargetDieLayout);
s_max ← ⌊S_MaxSize / TW_DiePitch⌋ ;               /* 20/2 = 10 */
s_opt ← 1 ;                                  /* save optimal cost */
c_opt ← ∞;
for s = 1 to s_max do
    M_SourceStamp ← buildSourceStamp(M_StampTile, s);
    M_TargetStamp ← buildTargetStamp(s);
    n_TargetDiesPerStamp ← s^2 ;                  /* squared stamp */
    u_SourceWafer ← getUtilization(M_SW, M_SourceStamp);
    u_TargetWafer ← getUtilization(M_TW, M_TargetStamp);
    cost ← calculateCosts(...) ;                   /* see Sect. IV */
    if cost ≤ c_opt then
        c_opt ← cost;
        s_opt ← s;
    end
end
```

size of 20 mm × 20 mm. A valid stamp layout option has to cover an integral count of target dies. Hence, 10 different stamp variants are possible (squares, where side lengths are integer multiples of the target die pitch, see Fig. 7c). This enables an exhaustive search within the parameter space to optimize the manufacturing costs (cost model in Sect. IV). Algorithm 1 shows the pseudo code for the cost optimization example.

Note that in this scenario the exhaustive search is a viable optimization approach due to the small solution space. The inclusion of further optimization parameters (e.g., layout of the target die) would require different methods, such as a feasible metaheuristic (e.g., simulated annealing).

Figure 8 illustrates the result of the calculations. The cost minimum around a stamp size of 200 mm^2 corresponds to a side length of 14 mm (the seven-fold of the target die pitch). Additionally, the stamp duration and wafer utilization curves are plotted. The stamp duration exhibits an asymptotic curve progression and drops quickly due to the reciprocal dependency on the stamp size. Thus, the possible cost savings by a further increase of the stamp is limited after reaching a certain stamp duration. On the other hand, the utilization of the stamp and target wafer is steadily decreasing due to the growing stamp size, causing an increase of the costs. Eventually, this leads to a renewed increase in *unit* costs, indicating the optimization potential of the stamp size.

VI. Application and Outlook

Conventional chip/package co-design optimization strategies have been focused so far on a single integration technology. Hence, they are not suitable for design problems that consider different assembly variants due to the lack of models of the assembly processes integrated into the design tools. The presented model is a first step towards such an optimization of a truly heterogeneous integration which we expect will become more dominant in the post-Moore era.

In the future, we will employ this new approach in an industrial co-design flow in order to find optimized die dimensions, source and target wafer layouts, and stamp designs with regard to manufacturing cost. Furthermore, we plan to extent our tool beyond the µTP technology to optimize other heterogeneous integration processes as well.

Acknowledgement

This work has received funding from the European Union's H2020 Programme (ECSEL) under grant agreement number 737465, from the BMBF of Germany and from the SMWA of the Free State of Saxony under grant agreement number 16ESE0231S (project "Microprince").

References

[1] A. Fontanelli, "System-in-package technology: Opportunities and challenges," in *9th International Symposium on Quality Electronic Design (ISQED)*, March 2008, pp. 589–593, https://doi.org/10.1109/ISQED.2008.4479803.

[2] R. Weerasekera, D. Pamunuwa *et al.*, "Two-dimensional and three-dimensional integration of heterogeneous electronic systems under cost, performance, and technological constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 8, pp. 1237–1250, Aug. 2009, https://doi.org/10.1109/TCAD.2009.2021734.

[3] R. Fischbach, J. Lienig, and T. Meister, "From 3D circuit technologies and data structures to interconnect prediction," in *Proceedings of the 11th International Workshop on System Level Interconnect Prediction (SLIP)*, 2009, pp. 77–84, https://doi.org/10.1145/1572471.1572485.

[4] T. Horst, R. Fischbach, and J. Lienig, "Design methodologies and co-design options for novel 3D technologies," in *Proc. of ANALOG 2018*, Sep. 2018, pp. 181–186, https://ieeexplore.ieee.org/document/8576854.

[5] R. Fischbach, T. Horst, and J. Lienig, "Assembly-related chip/package co-design of heterogeneous systems manufactured by micro-transfer printing," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2019, pp. 956–959, https://doi.org/10.23919/DATE.2019.8714884.

[6] C. A. Bower, E. Menard, and P. E. Garrou, "Transfer printing: An approach for massively parallel assembly of microscale devices," in *Proc. 58th Electronic Components and Technology Conf*, May 2008, pp. 1105–1109, https://doi.org/10.1109/ECTC.2008.4550113.

[7] C. A. Bower, M. A. Meitl *et al.*, "Heterogeneous integration of microscale compound semiconductor devices by micro-transfer-printing," in *Proc. IEEE 65th Electronic Components and Technology Conf. (ECTC)*, May 2015, pp. 963–967, https://doi.org/10.1109/ECTC.2015.7159711.

[8] R. Lerner, S. Eisenbrandt *et al.*, "Heterogeneous integration of microscale gallium nitride transistors by micro-transfer-printing," in *Proc. IEEE 66th Electronic Components and Technology Conf. (ECTC)*, May 2016, pp. 1186–1189, https://doi.org/10.1109/ECTC.2016.373.

[9] D. Gomez, T. Moore *et al.*, "Manufacturing capability of micro-transfer printing," in *Smart Systems Integration; 13th International Conference and Exhibition on Integration Issues of Miniaturized Systems*, April 2019, pp. 1–4, https://doi.org/10.5281/zenodo.2654512.

[10] S. Lamm, P. Sanders *et al.*, "Finding near-optimal independent sets at scale," *Journal of Heuristics*, vol. 23, no. 4, pp. 207–229, Aug. 2017, https://doi.org/10.1007/s10732-017-9337-x.

[11] J. Lienig and J. P. Cohoon, "Genetic algorithms applied to the physical design of vlsi circuits: A survey," in *Parallel Problem Solving from Nature — PPSN IV*, 1996, pp. 839–848, https://doi.org/10.1007/3-540-61723-X_1047.

[12] S. Wicht, U. Krieger *et al.*, "MICROPRINCE: Open access pilot line for micro-transfer-printing of functional components on wafer level," in *Smart Systems Integration; 13th International Conference and Exhibition on Integration Issues of Miniaturized Systems*, April 2019, pp. 1–2, https://doi.org/10.5281/zenodo.2654502.

[13] "MICROPRINCE – pilot line for micro-transfer-printing of functional components on wafer level," https://microprince.eu/, accessed: 2019-09-16.