Design Model and Data Management for 3D IC Development

Armin Grünewald, Kai Hahn, and Rainer Brück

***Abstract*—The development of 3D systems is a highly complex procedure. Next to a huge variation of possibilities on how to vertically integrate two or more dies, a lot of aspects regarding cost, design and application specific selection of technology have to be considered. Therefore a design model will be presented, that considers the mutual influence of design and process technology during the integration flow development. The design model has been implemented in a software system. To be able to develop integration flows, complex technology information is needed, which results in the necessity of a data management with usage of a technology database.**

***Index Terms*—3D integration, data management, design model, process flow**

I. Introduction

A promising solution for continuing the increase in complexity of VLSI described by Moore's Law is the stacking of integrated circuits. In the last decades this growth has been achieved by scaling the on-chip feature size. The scaling cannot continue forever, because of physical limits in the shrinking process and increased manufacturing cost. Therefore the so called three-dimensional (3D) integration technology is more and more emerging. During development of a 3D chip, due to the large number of possible processes, it is necessary to take into account restrictions given by fabrication and technology. Additionally, the choice of the integration flow is affected by product specific constraints. This implies the need to explore technology as well as design options, as has already been demanded in [1]. Furthermore [2] points out that a "co-optimization" of the system design and the system hardware is necessary.

This paper follows these considerations, but focuses on the choice of processes during the development of a 3D integration flow and the management of the required material and process data. Integration flow in this context stands for a specific sequence of integration process steps to combine a number of dies to form a 3D stack. Therefore a design model, which makes use of the mutual influence of design and process technology, will be presented. Additionally a prototype of a design flow generator with a dedicated data management is shown. Before this the technology background of 3D integration and 3D integration flows are given. This paper closes with a conclusion and an outline of future work.

The work presented in this paper is part of the EDA cluster research project NEEDS (project label 01M3090), which is funded by the German Federal Ministry of Education and Research (BMBF).

Armin Grünewald, Kai Hahn and Rainer Brück are with the Institute of Microsystem Engineering, Department of Electrical Engineering and Computer Science, University of Siegen, Hölderlinstraße 3, 57068 Siegen, Germany. E-mail: armin.gruenewald@uni-siegen.de

II. 3D Technology Background

A. Integration Technology

Seen from the process view, 3D integration can be outlined by the following three classes of technology:

1. TSV fabrication (Insertion of vertical connections into the die; TSV = Through Silicon Via)
2. Wafer thinning (Reduction of the thickness of the die to be able to contact the TSVs)
3. Wafer bonding (Connection of two wafers or dies)

While creating TSVs it is necessary to consider, at which point of time with regard to IC-processes and integration process steps the fabrication of TSVs takes place. Creating TSVs before CMOS processing (FEOL = Front-end-of-line) is called "via first", after FEOL but before BEOL (= Back-end-of-line) "via middle". A via formation after the complete IC process (Post-BEOL) is called "via last". The advantage of the latter option is that the IC wafer can be manufactured in a foundry which does not support TSV formation yet. The TSVs can then be added later by another foundry. Creating TSVs after wafer bonding ("via after bonding") is another possibility. Examples of processes are the "Bosch-process" (DRIE = Deep Reactive Ion Etching) to create the trench and Chemical Vapor Deposition (CVD) to fill the trench with copper or tungsten. In order to reduce the thickness of the die it is necessary to perform wafer thinning. Therefore depending on the choice of integration flow two options exist: The IC wafer will be either bonded directly on the 3D IC stack with the backside up and then thinned or temporarily bonded on a wafer handle and thinned.

For the bonding process the bonding objects as well as the orientation are of importance: The two main approaches to bond a wafer stack are "Wafer-to-Wafer" (W2W) and "Die-to-Wafer" (D2W). Since W2W stacking means to combine two complete wafers, this method is only suitable if all dies have the same size. Another drawback is the yield of the die stack, because it derives from the yields of the individual dies. If for example two wafers reaching a yield of 90% each, a stacking results in an 81% yield (yield loss through the stacking process is not considered).

D2W stacking facilitates the use of "known good dies" (KGD) so that a higher yield can be achieved. Additionally it is possible to have different die sizes. The orientation of the wafers/dies can be either "Face-to-Face" (F2F), "Back-to-Face" (B2F) or "Back-to-Back" (B2B) [3].

B. *Integration Flows*

In order to achieve a stacking of only two dies there exist nine possible flows to combine the integration processes with the FEOL and BEOL [3] [4]. Fig. 1 shows one example of the "Via last" approach: After the FEOL and BEOL processes the wafer will be temporarily bonded on a wafer handle and then thinned. After that the TSVs will be fabricated from the backside and the wafer will be finally bonded B2F to the other wafer.

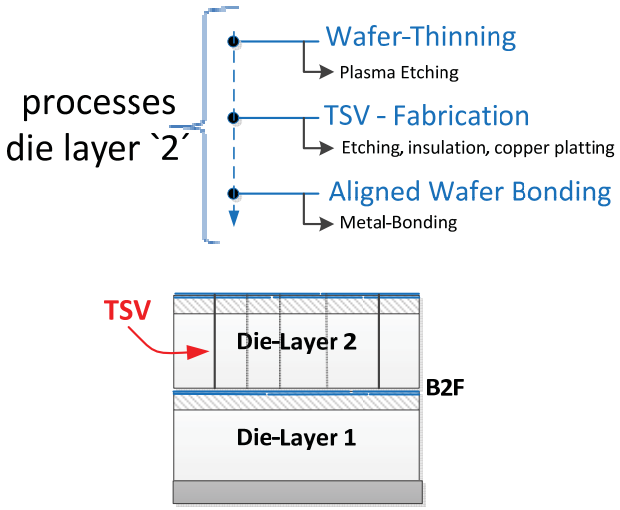

Figure 1. Example process flow.

With the large number of specific processes (e.g. DRIE or laser ablation for TSV fabrication) and materials to use, there are still a lot of possible variations after selecting one of the integration schemes. This accumulates even more when thinking about heterogeneous integration of several dies having a different technology each. In this case it will be very difficult to determine a process sequence which is fulfilling the initial specifications and is also manufacturable. Product specific constraints are for instance the number of dies, the technology of the dies, or the number of TSVs needed. These constraints will affect the selection of processes and TSVs. On the other hand pre- and post-conditions of different process steps, temperature budget and a cost factor need to be considered while developing an integration flow. These aspects can require major changes in the initial design. This train of thought leads to a design model approach, which is described in the following section.

III. Methodology For Creating 3D Process Flows

The foundation of the design model presented in this paper is the Pretzel Model developed at the University of Siegen (Fig. 2) [5]. The Pretzel Model originally designed for MEMS, is now adapted to 3D systems.

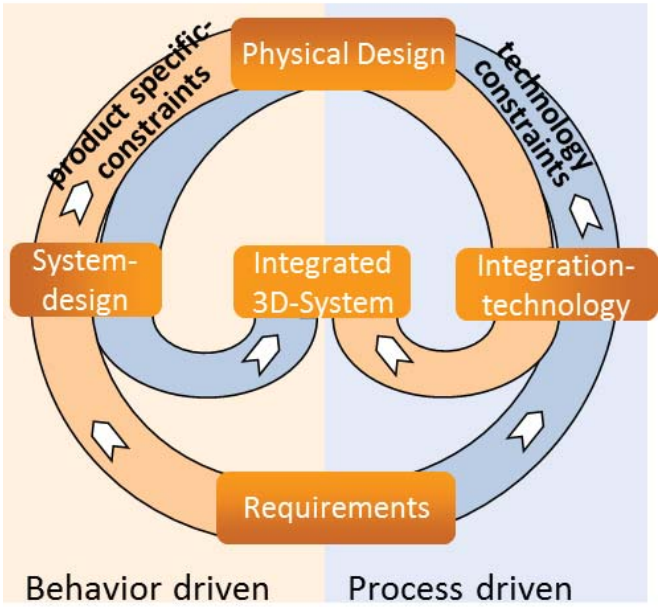

Figure 2. Pretzel Model.

In order to design an integrated 3D system, the bottom-up process design flow (right hand side) has to be taken into account at the same time as the top-down product design flow (left hand side), since with the high amount of dependencies as described before the design cannot be performed detached from the technology. According to this model the designer starts creating a structural description, which is based on the requirements. From this structural description a corresponding physical design of the 3D system can be derived. The next step is to design a process flow for manufacturing the 3D system. At the same time important material and process step data is collected, and the process flow has to be verified with this additional data. The last two steps have to be repeated until the process flow matches the physical design of the 3D system.

In order to be able to develop a process flow considering design and technology constraints at the same time, a design model has been developed. The design model is divided into four parts (Fig. 3), which are described in the subsequent sections.

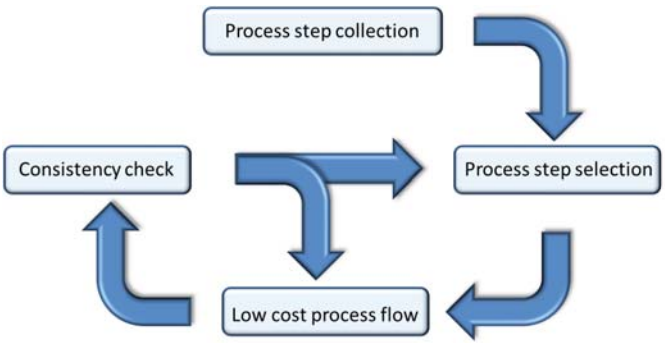

Figure 3. Methodology for creating integration flows.

A. Building of an Integration Process Step Collection

The first step to develop an integration process flow is to build up a collection of available process steps. Each process step needs some parameterization, so that different analysis on process steps and combination of process steps respectively process flows can be performed. Process parameters basically include information on temperature, processing time and involved materials as well as other constraints.

One crucial aspect is the cost factor (see section C). Cost is always an important challenge and therefore it is necessary that every process step is parameterized with a cost value, so that not only technical but also economic conditions can be considered during the development of the process flow.

Process steps require also different temperatures and too high temperatures can damage existing layers on the die. Therefore it is important to know the process step's contribution to the overall temperature budget.

Additionally every process step needs a list of pre- and post-conditions to indicate the dependencies to other process steps. With these conditions it is possible to perform a final check of the flow at the end of the development process to guarantee the manufacturability.

B. Reduction of the Number of Process Steps Due to the Consideration of Product Specific Constraints

After building an integration process step collection with the mentioned parameters it is necessary to take account of product specific constraints. These constraints give information of the desired product and are derived from the design exploration. They affect the selection of process steps and therefore allow building an integration flow which suits best for the product.

Possible constraints are the number of dies, technology of the dies and number of TSVs. The conditions of the different process steps therefore need to be checked on the individual inputs. In case of violations, the process step has to be removed from the list.

C. Selection of an Integration Flow with Preferably Low Cost Factor

The next step is to build a low cost integration flow from the remaining integration process steps. In recent research papers different approaches have been presented to perform a 3D IC cost analysis. In [6] a cost model for a 3D-System-on-Chip is presented where the cost of a 3D process step is determined by the processing time and material consumption per step. Out of these it is possible to derive yearly production cost while considering a yearly target production volume. The cost consists of equipment cost, clean room cost, personnel cost, maintenance cost and material cost. In comparison to this in [7] a cost model is presented where the final 3D chip cost is separated into several different parts: A wafer cost model takes account of the die area and the yield, and a bonding cost model determines the costs of the integration process steps. Both are merged into the overall 3D cost model, which also depends on additional design options such as Die-to-Wafer/Wafer-to-Wafer bonding or Known-Good-Die cost.

The cost model used for the design model includes the ideas presented in [6], but focuses on the costs of expensive materials, equipment/machine cost, and the processing time.

In case it is not possible to develop a low-cost integration flow, e.g. due to lack of available process steps, the building process is canceled and the violating constraints are listed. In this case it is necessary to go back and change the process steps / process parameters or the product specific constraints.

D. Consistency Check

The last step of the design model is a consistency check of the selected process to guarantee manufacturability. The technology constraints of the process flow and the individual process steps are represented by the pre- and post-conditions of each process step, which have been defined in section A. If for example for TSV formation a copper (Cu) deposition process step exists, it may have a deposition of an adhesion layer like TiN as a pre-condition and a thermal treatment as a post-condition [3].

These conditions allow to prove, if all necessary process steps are part of the flow and to directly indicate errors by showing the violating dependency. Another crucial aspect for a sequence of process steps is the temperature budget. Some process steps require a certain temperature limitation for the following process steps or the applied materials can only withstand a certain temperature over a particular period of time. In these cases it is necessary to analyze the different requirements of every process step and every material. In case of violations a return to step B or even to step A (then going along with changes in the design) is necessary, which means that it is also possible that no suitable flow exists.

IV. Integration Process Flow Modeling

In order to be able to describe a process step sequence and implement the description in a software system afterwards, a process graph model (PGM) has been developed. At first glance a list-like data structure seems appropriate, since a process flow is a sequence of individual processes. A closer examination reveals that this is not sufficient, because 3D integration technology involves at least two different dies or wafers, which leads to the necessity of representing more than one process sequence. Thus a graph was chosen as a data structure for its flexibility and extensibility. A simplified abstract example of a PGM is shown in Fig. 4. In the PGM a node represents the current "production state" of a die and an edge represents an integration process step (IPS). After performing an IPS (e.g. an etch process for via formation), the production state of the die changes, reaching another node in the graph. For developing a process flow of a die stack with three dies as given in the example in Fig. 4. Initially each die is represented by its own node and processed individually, until a bonding process occurs. In this case two different process sequences are merged together to create a new node (e.g. "Die 1+2 – state 0)", which represents the newly stacked die stack. The PGM is presented in more detail in [8].

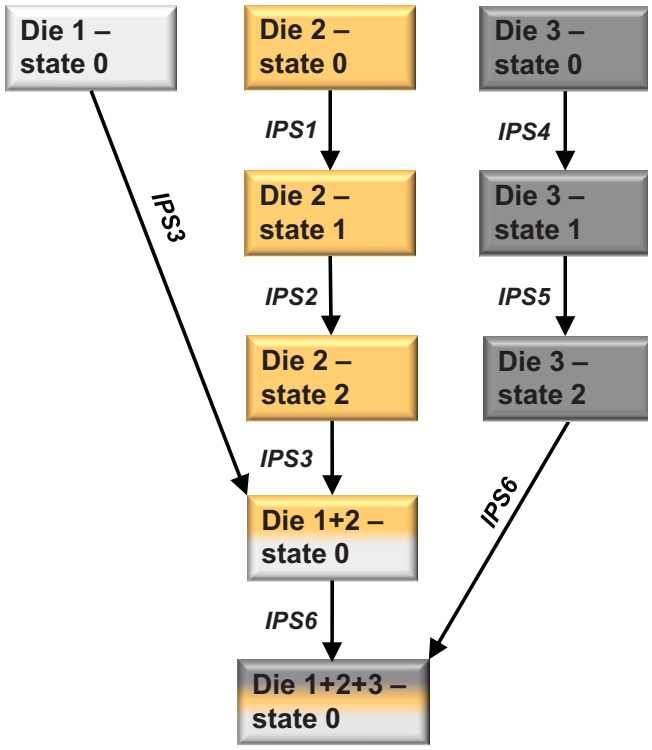

Figure 4. Example of a process graph model.

V. Data Management for Product Specific Integration Technology

Developing 3D systems is a highly complex procedure, since a huge amount of different technology data has to be considered. Therefore it is necessary to be able to create, edit and manage data in a comfortable way and also export the data in a format, so that the information can be easily retrieved by other design tools.

A. Database

The requirement for the database is to store all data that is used for the integration flow generation as well as for other tools regarding the design process of a 3D system. This is achieved by starting with basic information first. To describe technological and physical characteristics, units and parameters are essential components. The data stored in the database to specify a unit are the name, a unique ID, a unit symbol, a short description and information about the author and the creation date. A unit has no dependencies to other entities in the database, so that the data can be retrieved with a simple SQL command "Select * from unit". A parameter is described in a similar way. Additional information is the assigned unit, the type of the value (numerical, textual or material) and attributes. A parameter is initially abstract, which means that it has no attached value. The value will be set when the parameter is assigned to a material, process step, layer or TSV. Attributes adds further information to a parameter, e.g. to specify a location where a certain temperature has to be reached. With the dependencies to attributes and unit it is no longer possible to extract data from the database or insert new data with a simple SQL command. Therefore a data editor has been developed, which is described in the following section.

The most important data stored in the database are information about materials, TSVs, dies, process steps and process flows. Materials are described with a name, a unique ID, a description, information about the author and creation date, a list of parameters and their values and the material class it belongs to. While TSVs are described in a similar way, a process step and a die need some additional information. The parameters of a process step are divided into process and result parameters in order to give a more accurate specification of a process step. To be able to perform a consistency check over the whole process flow, every process step and die is parameterized with rule sets. For each process step individual pre- and post-conditions are defined to describe the position of the process step inside a process flow. The process flows stored in the database are a mapping of the process graph model mentioned in the previous section. Additionally each die can be parameterized with conditions, e.g. to prevent the use of high-temperature process steps for a die with a certain technology.

The different data objects are not isolated structures, but are strongly interrelated. An overview of the data objects and their (inter-)dependencies is shown in TABLE I. Taking for instance a look at the process steps, dependencies to TSV, die, material and parameter can be found. In order to be able to describe a process step properly, it is necessary to define process parameters and involved materials. While choosing a suitable process step, the technology of the die has to be considered as seen before. The selection of process steps for via formation also affects the achievable geometrical dimensions of the TSVs.

TABLE I
Data Objects and Dependencies in the Technology Database

	dependencies	example data
material	parameter	copper, tungsten, silicon, …
process steps	TSV, die, material, parameter	DRIE, PECVD, grinding, …
TSV	material, parameter	Copper 15 µm, …
die	process step, TSV, parameter	CMOS 65 nm, …
process flow	process steps, TSV, die	Demonstrator_Flow

B. Data Editor

As described before, the data structure of the different objects is complex, especially for materials and process steps. To be able to add and edit data easily, the client of the software ASPIRE (see section VI) has different editors with a

graphical user interface. Fig. 5 shows a screenshot of the material editor together with the popup for editing a material parameter. In the given example the material copper has two parameters (thermal conductivity and density). The density has the value 8920 kg/m³. Except for the process flow editor all other editors are built up in a similar way. The process flow editor uses a graph model to illustrate a process flow (section VI).

The inputs that are made in the GUI are translated into SQL commands by the software and send to the technology database, where the according data set is created or updated. These tasks are performed invisibly in the background, so that no further action from the user is required.

C. Exchange Format

Next to the functionality to manually add and edit technology data as described in the previous subsection, it is necessary to import and export data in order to communicate with other design tools. As a text format XML (Extensible Markup Language) [10] has been chosen, because of its advantages such as being platform-independent, extendable, flexible and easy to integrate in applications with the help of a parser (e.g. JDOM for JAVA). Fig. 6 exemplary shows the XML export of the material copper (already shown in the editor window in Fig. 5). In the XML-file it is also possible to see the creation date, the ID of the author and the ID of the material. Each data object in the database has a unique ID to be clearly identifiable, which also needed for the (inter-) dependencies between the different data objects.

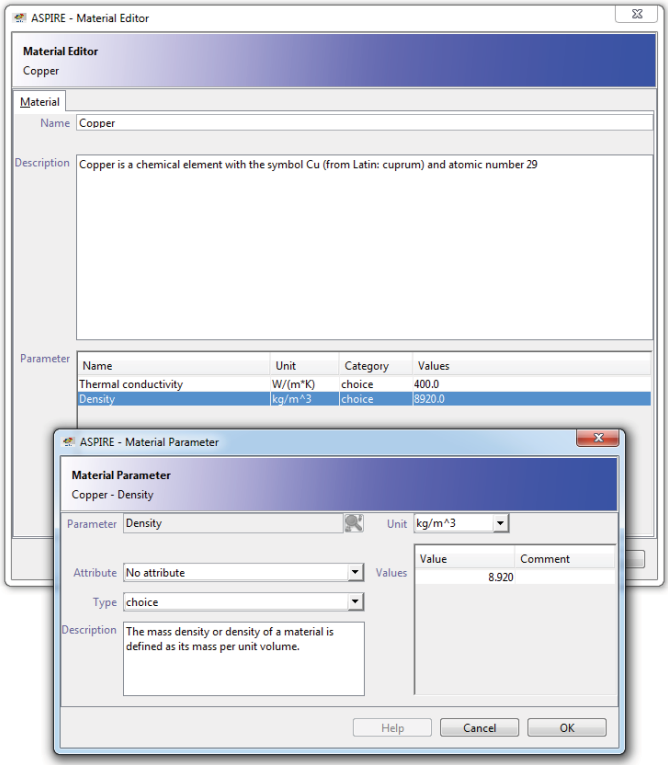

Figure 5. ASPIRE material editor.

```
<?xml version="1.0" encoding="UTF-8"?>
<database>
 <material>
  <name>Copper</name>
  <materialID>45</materialID>
  <description>Copper is a chemical element with the
       symbol Cu (from Latin:cuprum) and atomic
       number 29.
  </description>
  <authorID>6</authorID>
  <creationDate>2011-12-01T12:21:26Z</creationDate>
  <parameter>
   <name>Density</name>
   <description>The mass density or density of a material
       is defined as its mass per unit volume.
   </description>
   <unit>weight per volume</unit>
   <type>choice</type>
   <value>8920.0</value>
  </parameter>
  <parameter>
   <name>Thermal conductivity</name>
   <description>In physics, thermal conductivity, k, is the
       property of a material's ability to conduct heat.
   </description>
   <unit>Watts per meter kelvin</unit>
   <type>choice</type>
   <value>400.0</value>
  </parameter>
  <materialclass>Metal</materialclass>
 </material>
</database>
```

Figure 6. XML Description of material copper.

VI. System for 3D Integration Flow Generation

A. Overview

The described design model is implemented in the software ASPIRE (“Application Specific Integration Flow Evolution”) (Fig. 7). It is realized as a client-server architecture in combination with a PostgreSQL [9] database, which has already been introduced in the previous section. With the help of the client it is possible to add and edit process steps, dies, materials and TSVs. After creation of this basic data (section V) it is possible to develop process flows with a process flow editor. In order to be able to create integration process flows, that allow the assignment of process steps to different dies, a dedicated data model for describing process flows has been developed (section IV). In the following subsection the process flow editor is presented. In order to be able to check the feasibility of the developed process flows, a consistency check has been implemented, which is also shown with an example.

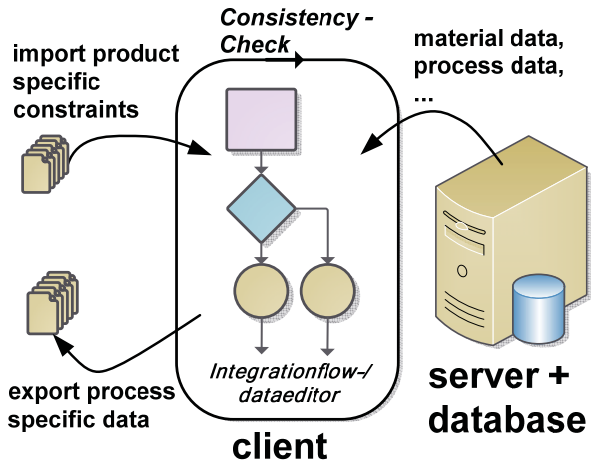

Figure 7. Overview software system.

B. *Flow Editor and Consistency Check*

An illustration of the process graph in ASPIRE, which is an implementation of the PGM as described in section IV, is shown in Fig. 8. The example in the Process Graph Editor Window shows a process step sequence of "Die B". Every time a manipulation of the graph in the GUI is performed (e.g. a node or edge is added), the change is transferred to the underlying process graph, which is directly connected to the database. This allows the storing and loading of process flows. A visualization of a stored process flow is done by using a depth-first search (DFS) on the process flow stored in the database. During the development of the process flow the user is supported by the software. By describing the predefined die-stack in XML format (with information about the number of dies, technology and alignment of the dies) the editor starts with the according number of die states and establishes a connection to the according dies in the database. When the user adds an integration process step to the process flow, the process steps are checked against the conditions of the involved dies. This prevents the use of process steps that are not compatible to a specific technology. As described before, each process step can be parameterized with conditions, e.g. to define which process steps has to be taken before and after the current process step.

In the given example the process step "DC-Sputtering" has the pre-condition that the process step PECVD must be performed in advance. This condition is not fulfilled. Since all process steps are connected in a graph, this violation can easily be detected by a consistency check. The consistency check, implemented to assist the user in creating a manufacturable process flow, tests every condition of every process step and marks discrepancies. The result of the check is shown in a popup window (Fig. 8). Additionally limitations given by the technology of the die or the choice of TSV are also checked.

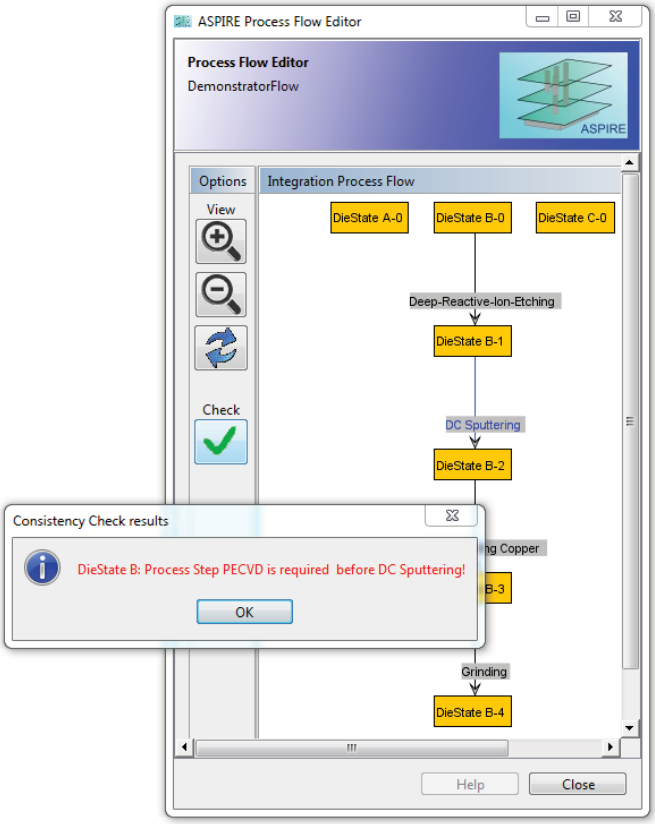

Figure 8. Process graph editor and consistency check

VII. Conclusion and Future Work

A design model for developing application specific 3D integration flows accompanied by a process design software prototype with a dedicated database were presented in this paper. With the design model it is possible to take account of both process and product specific constraints. Provided that 3D integration is feasible, the software described in the two previous sections allows the development of manufacturable process flows out of a set of available integration process steps. Currently editors for materials, TSVs, dies, and process steps exist as well as the connection to a database which allows the storage of all necessary technology data. Additionally it is possible to develop 3D integration flows using a process flow editor, which is an implementation of the presented process graph model. With the help of a consistency check the process flow can be checked on feasibility at any time. Furthermore, an implementation of a cost model as described in section III is under development so that economic aspects are also considered. A light-weight integration with other design tools to be developed within the project NEEDS (e.g. 3D floor planner, 3D test) is planned.

REFERENCES

[1] P. Marchal et al., "3-D Technology Assessment: Path-Finding the Technology/Design Sweet-Spot," Proceedings of the IEEE, vol. 97, 2009.

[2] P. Emma, E. Kursun, "3D system design: A case for building customized modular systems in 3D," Interconnect Technology Conference (IITC), 2010.

[3] P. Garrou, C. Bower and P. Ramm, "Handbook of 3D Integration," Wiley-VCH, 2008.

[4] Yole Développement, "Thin wafer handling for 3D TSV," SEMATECH Workshop SEMICON West, 2011.

[5] K. Hahn, A. Wagener, J. Popp and R. Brück, "Process management and design for MEMS and microelectronics technologies," Proceedings of SPIE: Microelectronics: Design, Technology, and Packaging, vol. 5274, 2003.

[6] D. Velenis, M. Stucchi, E.J. Marinissen, B. Swinnen and E. Beyne, "Impact of 3D design choices on manufactering cost," IEEE International Conference on 3D System Integration, 2009.

[7] X. Dong and Y. Xie, "System-Level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs)," Asia and SouthPacific Design Automation Conference, 2009.

[8] A. Grünewald, K. Hahn, R. Brück, "Software-based Development of 3D Integration Flows," 35th International Electronics Manufacturing Technology Conference, 2012, in press.

[9] PostgreSQL: http://www.postgresql.org/

[10] XML: http://www.w3.org/XML/

Armin Grünewald was born in Altenhundem, Germany in 1985. From 2005 to 2010 he studied applied computer science with application in electrical engineering at University of Siegen, Germany. He received his Diploma in computer science from the department of computer science and electrical engineering of the University of Siegen in 2010.
Since the beginning of 2011, he works as a Research Assistant at the Institute of Microsystem Engineering at the University of Siegen. His current research interests are 3D integration in the NEEDS project and design and implementation of integrated circuits and microsystems.

Kai Hahn was born in Dortmund, Germany in 1961.
From 1982 to 1990 he studied electrical engineering at Dortmund University, Germany. He received a doctoral degree (Dr.-Ing.) in computer science from Siegen University in 1998. From 1990 to 1991 he worked as a CAD engineer in software company (DOSIS GmbH). From 1991 to 1998 he worked as a researcher at the computer science department of Dortmund university. His research focus was initially in the area of electronic design automation and shifted to the field of MEMS design and design tools. Since 1998 he is working as a senior researcher at the electrical engineering and computer science department of University of Siegen, Germany. His current research focus are methods and tools for interfacing the design and the manufacturing of micro and nano products as well as product engineering for MEMS and IC systems. Dr. Hahn is a member of VDE and VDI.

Rainer Brück was born in Borken, Germany in 1958. From 1977 to 1983 he studied computer science at Dortmund University, Germany. He received a doctoral degree (Dr. rer. nat.) in computer science from Dortmund University in 1989. In 1996 he received the venia legendi in computer science.
He worked as a research assistant from 1983 to 1989 and as an assistant professor from 1989 – 1996, both at the computer science department of Dortmund university. His research was in the field of electronic design automation. From 1986 to 1996 he headed the physical research group, where the first commercially successful analog layout compaction system CAMBIO-XT was realized. Since 1993 the focus of his research shifted to the field of MEMS design and design tools. From 1996 to 1998 he worked as an associate professor at the mathematics and computer science department at Friedrich-Schiller-Universität Jena, Germany. Since 1998 he is working as a Full Professor at the electrical engineering and computer science department of University of Siegen, Germany. His current research focus is in computer aided product engineering for MEMS and IC systems.
Prof. Brück is a member of GI, VDE and VDI.