From: AAAI-86 Proceedings. Copyright ©1986, AAAI (www.aaai.org). All rights reserved.

Qualitative Simulation of Semiconductor Fabrication

John Mohammed

Schlumberger Palo Alto Research
3340 Hillview Avenue
Palo Alto, CA 94304

Reid Simmons

MIT AI Laboratory
545 Technology Square
Cambridge, MA 02139

ABSTRACT

As part of a larger effort aimed at providing symbolic, computer-aided tools for semiconductor fabrication experts, we have developed qualitative models of the operations performed during semiconductor manufacture. By qualitatively simulating a sequence of these models we generate a description of how a wafer is affected by the operations. This description encodes the entire history of processing for the wafer and causally relates the attributes that describe the structures on the wafer to the processing operations responsible for creating those structures. These causal relationships can be used to support many reasoning tasks in the semiconductor fabrication domain, including synthesis of new recipes, and diagnosis of failures in operating fabrication lines.

I Introduction

Semiconductor fabrication is the long and complex process by which wafers of almost pure crystalline silicon are turned into integrated circuits. It is carried out according to a *recipe*, which is a linear sequence of parameterised operations that defines how to create devices belonging to a particular technological family such as Bipolar, NMOS or CMOS.

The work described in this paper is part of a larger effort aimed at providing computer tools to facilitate diagnosis and the design of process recipes. In this paper we focus on the development of qualitative models which are used to reason symbolically about the fabrication process.

The scenario we envision is shown in Figure 1. The "generic knowledge-base" would contain models of the processing operations used in fabrication, such as "etching" and "oxidation." It would also include models of the electronic behaviour of the devices being fabricated, and models of the manufacturing equipment used. A suite of symbolic reasoning tools would use these models to help the process designer create a recipe for a new process. The result of this design process would be a "recipe-specific knowledge-base" containing all the knowledge gained about the recipe and about the fabrication process it represents. Computer tools utilizing both the general knowledge and the recipe-specific knowledge would aid the production engineer in his tasks of improving the yield of the process and diagnosing failures.

Today, the primary computer tools available to process designers are numerical, incremental-time simulators (e.g. [Ho and Hansen]). These simulators use mathematical models of the physical and chemical processes employed in semiconductor fabrication to determine the results of applying a recipe to a prototypical wafer. Such simulators do provide a very important source of quantitative information that might otherwise be obtained only by performing costly experiments with real wafers.

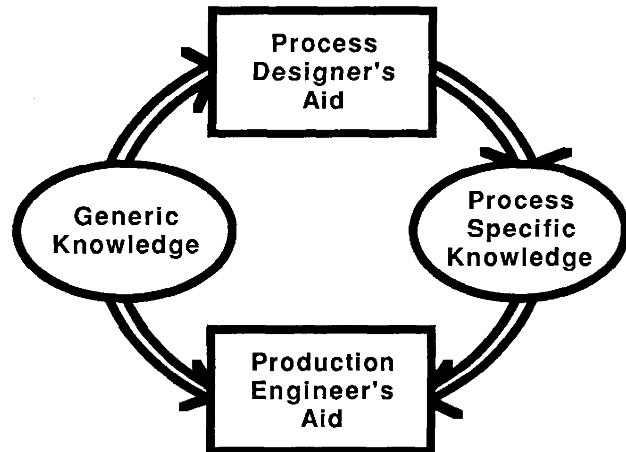


Figure 1: General scenario for Semiconductor Fabrication CAD/CAM tools.

However, process designers and production engineers do much causal reasoning about the fabrication process for which numerical simulators provide little or no aid. This reasoning typically involves relating attributes of the wafer to operations of the recipe. For example, when the resistance of some layer on the wafer is found to be too high, an engineer might want to know which operations might have been responsible. Also, the process designer or engineer often needs only a qualitative answer to a partially specified question, such as "will the resistance of layer X increase if the temperature of step 5 is increased?".

In order to automate this type of reasoning, we have constructed qualitative, causal models for each type of fabrication operation. Each model describes how the structure of a wafer is affected by an operation. We have chosen to model operations at a level that captures the process engineer's "naive" understanding of semiconductor manufacturing. This level is sufficient for many of the causal reasoning tasks an engineer would want to perform, yet it suppresses the unnecessary detail and mathematical sophistication that are required for accurate numerical simulation.

These models constitute a set of "building blocks" that can be strung together to form a recipe. Our simulator takes such a recipe as input and produces a *wafer history*. A wafer history describes how the structure of a prototypical wafer evolves over time as the fabrication processing proceeds. It also records causal dependencies that relate the structural attributes of the wafer to the operations responsible for generating those structures. This causal dependency information can be used to sup-

port diagnosis of failures on a running fab line and can help in the synthesis of new recipes.

We discuss our models in the next section. In Section III we then briefly describe the language in which the models are written, and the qualitative simulator. Section IV describes the reasoning tasks we have performed using these models, which include qualitatively simulating the fabrication of several devices according to a recipe for a bipolar process and using the causal dependency information gained to support diagnosis. Finally, we present several research issues related to this work.

II The Models

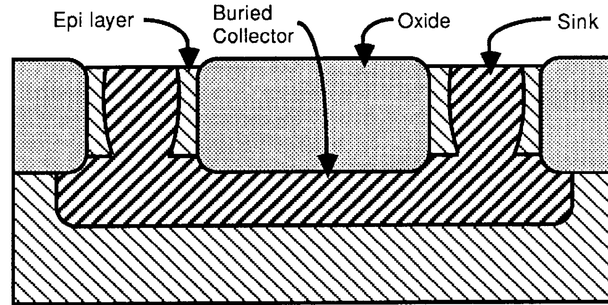
A. Modeling the Wafer

In semiconductor manufacture, electronic devices are formed on the "upper" surface of a thin wafer of silicon. A device is a three-dimensional structure with a particular geometric configuration of regions of silicon (possibly with controlled amounts of impurities embedded), silicon compounds and metals. This section describes how we model these structures by explicitly representing physical, topological and geometric attributes of the wafer.

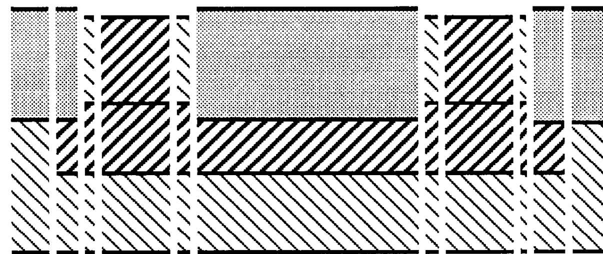
The qualitative reasoning techniques that have been developed in AI apply mainly to reasoning about scalar quantities related by partial orders. In order to employ these techniques we have adopted a simplified representation of wafer structure. Fortunately, much of the reasoning that fabrication experts do requires that only two-dimensional vertical cross sections of the wafer structures be represented. Furthermore, the cross section can be usefully modelled as a series of vertical strips (see Figure 2). Many of the numerical simulation tools (e.g. SUPREM [Ho and Hansen]) simplify the wafer representation in the same way. Thus, we can describe wafer geometry as essentially a one-dimensional horizontal series of one-dimensional vertical layers. We represent the horizontal axis of the cross-section by a series of horizontal *regions*, explicitly representing their lateral extents and lateral topology (i.e., left and right regions). Unlike many fabrication simulators, our simulator actually creates this lateral topology using a simple representation of photolithography masks. For each horizontal region, we represent the sequence of vertical *layers* one would encounter in going down through the wafer at a point in the interior of the horizontal region. Besides describing the vertical topology, a layer has attributes describing the material of the layer, any dopant and its concentration and the thickness of the layer.

With the exception of mask-exposure, all the processing steps are described as "vertical processes"—only their effect on the vertical geometry of the layers within each horizontal region is described. Our models ignore the effect of such operations on the transitions between adjacent horizontal regions. For example, during an etching step material is removed from the layer at the surface of the wafer in those areas of the wafer that are not protected by photoresist. Realistically, in some etching techniques the etchant can remove material in a lateral direction as well as vertically, and thus "encroach" upon an adjacent horizontal region. This lateral effect is not described in our model of the etching operation.

The one operation which cannot be specified as a "vertical process" is mask-exposure. However, mask-exposure is concerned exclusively with how masking affects lateral geometry on a region by region basis. Thus, modeling the wafer as a series horizontal regions is sufficient to capture the effects of masking.



a. Vertical cross section through a test structure (CUO)



b. Strips representing vertical topology within horizontal regions.

Figure 2: Representation of wafer cross-section as vertical strips.

We have found this to be a reasonable approximation for many of the reasoning tasks we wish to undertake. For example, in section IV we see how these models support some fairly detailed diagnostic reasoning.

B. Modeling the Operations

Structures are created on a wafer by the application of a recipe, which typically requires between 100 and 200 fabrication steps. However, all these steps are drawn from a comparatively small repertoire of standard parameterized operations. We have modelled a reasonably complete set of these operations. They can be grouped into categories as follows:

Addition of Material: these operations cover the upper surface of the wafer with a "blanket" layer of some material.

1. Chemical-Vapor-Deposition—deposits silicon compounds like silicon nitride and silicon dioxide;
2. Epitaxial-Growth—grows crystalline silicon;
3. Spin-On-Resist—coats the wafer with a positive or negative photoresist; and
4. Sputtering—deposits metal layers;

Removal of Material: these operations remove material from the upper surface of the wafer (selectively, based on material type).

1. Etch—(we do not distinguish between "wet" acid bath, or "dry" plasma etch) removes materials other than photoresist;
2. Photoresist Clean—removes all photoresist independent of "hardness"; and

3. Photoresist Develop—removes only "soft" photoresist;

Change of Chemical Properties: these operations modify the chemical composition of existing layers.

1. Mask-Expose—changes the "hardness" of a layer of photoresist by using light or X-ray radiation to break or form chemical bonds; the radiation is patterned with a mask; this is the primary method by which the surface of the wafer is differentiated laterally into distinct regions to form devices and wires; and
2. Oxidation—combines silicon and/or silicon compounds with oxygen to form silicon dioxide;

Change in Doping Profile: the controlled introduction of impurities into the silicon crystal lattice is the key to the formation of devices that have interesting electronic behaviour; these operations effect the presence and control the concentration of these impurities.

1. Diffusion—modifies the distribution of impurity ions by permitting them to diffuse through the crystal;
2. Ion Implantation—accelerates ions of an impurity electromagnetically towards the wafer to implant them to a depth determined by the energy imparted to the ions; and
3. Pre-Deposition—introduces impurity ions in very high concentrations at the surface of the wafer.

Each of these operations is parameterised. The parameters may be numeric or non-numeric. Numeric parameters specify, for example, the temperature at which an operation should occur. An example of a non-numeric parameter is one that specifies the particular etchant used in an etch operation.

A recipe consists of instances of these operations, with particular values specified for the parameters. As an alternative to specifying numbers for numeric parameters, our system permits qualitative constraints on the values of parameters to be specified. For example, rather than stating that the duration of an etch step is twenty minutes, one can state that it is "long enough to completely remove the uppermost layer." This is especially useful during the design of a new recipe, when the designer has in mind what the effect of the operation should be, but has not yet determined what values for the parameters are necessary to achieve that effect.

We represent the effects of an operation as a conjunction of logical implications. The consequents of these implications are the changes that occur to the world, including the creation and destruction of objects. The antecedents of the implications describe the conditions under which these changes occur.

As our model of the Etch operation is indicative of the nature and style of our models of processing operations, the rest of this section describes how we model that operation. Descriptions of all the models can be found in [Simmons and Mohammed].

Etching acts to remove material from the uppermost layers on the wafer, thereby reducing a layer's thickness or destroying it altogether. In the event that layers are completely consumed, the topology of the wafer changes and a previously buried layer becomes the new uppermost layer. The total amount of material removed depends on the *duration* of the operation and the particular *etchant* used, and may differ from one region of the wafer to the next due to the fact that etching occurs at different rates for different materials.

Thus, in order to determine the effect of an etch operation, our model must determine whether each layer in each horizontal region of the wafer is etched at all, and if so, how much is etched away. A layer is totally etched away if the *duration* of the operation is longer than the sum of two durations: (i) the time needed to etch through all layers above the layer; and (ii) the time needed to etch through the layer itself. We call this sum the *Etch-Destroy-Time* of the layer, and represent it as a function that depends on the layer's thickness, the *Etch-Rate* (another function) at which the *etchant* etches through the material of the layer and the *Etch-Destroy-Time* of the layer above it. A layer is partially etched away if the *duration* of the operation is shorter than the *Etch-Destroy-Time* of the layer but longer than the *Etch-Destroy-Time* of the layer above it. The amount by which the thickness of the layer is reduced is determined from the *Etch-Rate* of the *etchant* for the material type of the layer and the difference between the *duration* of the operation and the *Etch-Destroy-Time* of the layer above it.

The models of the other processes use techniques similar to those described above, and most are fairly good approximations to the actual fabrication operations.

The only real exceptions are the ion-implantantion and diffusion operations that deal with the distribution of impurities within the wafer. The models we have written are complex yet not very faithful to reality. This is because of the difficulty of representing concentration profiles in a way that the simulator can reason about them. We haven chosen a very simple way to model impurity profiles: within each layer there can only be one dopant and the concentration of that dopant is considered to be constant throughout the layer. Concentration profiles are thus modelled as simple combinations of step functions. Two vertically adjacent layers that are made of the same basic material may be distinguished by the concentrations of impurities within them.

III Representation

The nature of our fabrication models has an impact on the features required in the language used to write them. The language must be capable of describing the changes that occur to attributes of the wafer, such as the thickness or existence of layers. These changes are often complex functions of the attributes of the wafer before the operation and the parameters of the operation. The simulator must be able to reason qualitatively about such functions. Finally, the language must make it possible to say that the same effects occur conditionally to all the layers of the wafer.

The language we use to model the operations is an extension of the discrete action languages that have traditionally been used in the planning domain. An action is "discrete" in the sense that it maps the state of the world at the instant before the action occurs to the state of the world at the instant after the action occurs, but says nothing about the state of the world while the action is occurring. This type of model has a rich history in AI [Fikes and Nilsson, Sacerdoti, Stefik].

The language and the qualitative simulator were originally developed for doing geologic interpretation and are described in detail in [Simmons 83]. Briefly, the language extends the traditional precondition/effects representation in that it allows

1. Effects that are expressed in terms that are relative to the input state (e.g. "the thickness of layer L decreases by 5")
2. Effects that are universally quantified (e.g. "for all layers, the thickness decreases")
3. Effects that are conditionalized (e.g. "if the layer's material is silicon, then the thickness decreases")
4. Creation and destruction of objects.

Arithmetic functions can be used in the specification of the effects of an action. The simulator can reason about the value of a particular function application either from the definition of the function (if it is supplied) or from constraints on the possible values for the function. For example, our models include the definition of the *Etch-Destroy-Time* function described above, and thus the simulator knows that the value of the function for a layer depends recursively on the value of the same function for the layer above that layer. From this definition and the constraint (provided by the process designer) that the *duration* of the Etch operation is greater than the *Etch-Destroy-Time* of a given layer, it can determine that the *duration* is longer than the *Etch-Destroy-Time* of all layers above the given layer.[1]

Time is represented explicitly as point-like instants. Time intervals are defined by their end-points. One can assert ordinal relationships between time instants ($>$, $<$, $=$, $\geq$, $\leq$, $\neq$). The simulator maintains a consistent partial order and can deduce new relationships based on the transitivity of existing ones. Basically, these temporal relationships allow one to temporally order operations and to refer to the state of the world at different points in time. Use of a partial order permits one to store and reason about incomplete temporal information.

The "world model" is a set of objects. Like typical frame systems, the objects have a set of attributes and the object types form a simple inheritance hierarchy. The set of attributes for an object of a given type includes those of any superior type. Unlike typical frame systems, our world model includes a temporal dimension. First, objects have a temporal extent. Thus, we can talk about when an object was created or destroyed. Second, the "value" of an attribute is represented as a sequence of *intervals* called a history.[2] This sequence encodes the complete history of how the attribute's value changes over time. The intervals in the histories are of two types. "Dynamic" intervals indicate that some change was occurring to the attribute during that interval of time. "Quiescent" intervals indicate that no change was occurring and therefore the value of the attribute remained constant during the interval. The value associated with each interval is either a quantity (such as the thickness of a layer), another object (such as the neighbouring layer) or a set of objects. Quiescent intervals encode a non-monotonic *persistence assumption* about the world. All attributes of all objects are assumed to be quiescent (unchanging) during every time interval for which there is no evidence that their values are changing.

A. Recipes and the Simulator

A recipe is implemented simply as a list of "events." The first event is an initialization step to create objects representing the initial wafer structure, the materials, such as NITRIC-ACID, to be used in the fabrication and the various masks to be used in the recipe.

Each subsequent event represents a particular manufacturing step. The description of each event includes the type of the operation and a set of constraints. Typically, the constraints are assertions of qualitative relations between parameters of the operation and attributes of the wafer. In the absence of numerical information, these constraints enable the simulator to infer the extent of the effects of each event. For example, by asserting that the *duration* of an etch operation is less than the *Etch-Destroy-Time* of the layer at the surface of the wafer the simulator is able to infer that the top layer is only partially etched and that none of the layers below the top layer gets etched at all.

[1] This reasoning is accomplished using the "quantity lattice" described in [Simmons 86].

[2] Our notion of history is derived from, but not identical to, that of [Hayes].

The simulator works by *instantiating* each event in the list. The constraints and the definition of the model for the indicated type of operation enable the simulator to infer which changes occur to the world model. The simulator then manipulates the wafer history to reflect these changes. The end result is a set of objects whose attributes describe the complete history of how the object changed over time. This wafer history, which includes causal dependencies recorded by the simulator, essentially forms a causal explanation in terms of the events in the recipe and their parameters.

IV Experimental Results

We have successfully simulated the fabrication of several typical devices according to a recipe for an oxide-isolated bipolar process. Our representation of the recipe involves forty-eight steps, including six masking steps. This includes all the essential steps in the recipe through the addition of metal contacts. The only steps omitted are those preparatory steps which do not directly affect the topology or geometry of the wafer structure such as the gettering step, wafer cleaning steps, dehydration steps and photoresist baking steps.

We have implemented a capability that graphically displays the state of the wafer at each processing step in order to provide visual feedback concerning the progress of the simulation. Figure 3 is a sample of this output, showing a NPN bipolar transistor. In order to generate the coordinates needed to draw the display, the system determines symbolic expressions for the geometric attributes of the structures, such as the thickness of a layer. These expressions are obtained by tracing the dependencies through the wafer history and are given in terms of the parameters of the processing operations. They are then evaluated using approximate values for the parameters that are provided by the user.

The dependencies recorded by the simulator make it possible to determine which operations influenced an attribute of the wafer and how the value of the attribute depends functionally on the parameters of those operations. This ability to trace causal dependencies is an important component of several reasoning tasks, such as diagnosis of failures in processing on a production line and synthesis of new recipes. We describe our investigations into the role of this ability in the diagnostic task below.

In addition to the product circuits, a small number of devices called "test structures" are created on every wafer in order to facilitate quality control. The electronic properties of these devices are measured and when these measurements lie outside their expected ranges the wafers are rejected. The measurements then provide information helpful in diagnosing the problem.

Perturbations in the input parameters of processing operations form a useful fault model for many semiconductor manufacturing problems. Under this fault model, each of the input parameters that a measured attribute depends on gives rise to a diagnostic hypothesis for explaining an abnormality in the measurement: namely that the input parameter has an appropriately abnormal value itself. The expressions we obtain that relate the measurable attributes of the wafer to the input parameters of the operations can be used to order these perturbation hypotheses according to the sensitivity of the attribute to changes in each input parameter. Further, by tracing causal paths forward from each suspect input parameter to the attributes that they affect, one can determine what other measurements on the wafer would constitute confirmatory or contradictory evidence for the hypothesis.

Figure 4 represents two test structures placed side by side, called Collector-Under-Silicon (CUS) and Collector-Under-Oxide (CUO). The CUS structure extends between the first two sinks from the left. As its name suggests, in this structure the collector is below the silicon layer formed during an Epitaxial Growth step. The CUO structure extends between the second and third sinks, with most of the buried collector under an isolation oxide layer. For both devices the electronic property measured is the resistance between the sinks. In each case, the dominant influence on the measurement is the resistance of the buried collector, but the measurements also reflect the resistances of the sinks and (for CUS) the epitaxial layer.

We have implemented a program that uses the results of the simulator to identify all the input parameters that affect the resistance of each of these test-structures and to obtain expressions for the functional dependence of the resistance on these parameters. Since most input parameters contribute to the value of several measurable attributes on more than one test structure, the program can combine information from both normal and abnormal measurements to prune the set of fault hypotheses. First, those hypotheses concerning input parameters that contribute to measurements that are within their normal ranges can be eliminated as candidates. Second, those input parameters that would have to be abnormal in one direction to explain one measurement, and simultaneously abnormal in the other direction to explain another abnormal measurement can be eliminated. For example, the factors that control the resistance of the sinks affect both the CUS and CUO structures equally. Thus, a normal CUS measurement exonerates those factors as contributors to the abnormality of the CUO measurement.

An exception to this simple candidate elimination rule occurs when an input parameter plays a large role in determining the value of one measurement, but has only a negligible effect on the value of another one. For example, in the CUO structure, the narrow regions labelled "B" in Figure 4 undergo processing identical to that for the large part of the CUS structure labelled "A" in the figure. This means that every input parameter influencing the CUS resistance measurement also appears as a factor controlling the CUO measurement. By the simple rule discussed above, a normal CUO measurement would exonerate all the input parameters affecting the CUS measurement. However, in reality a normal CUO measurement can be consistent with an abnormal CUS measurement, because the factors governing the resistance of "B" have only a relatively small affect on the total CUO resistance.

This underlines the importance of considering sensitivity of the functional dependencies, and hence the importance of being able to generate symbolic expressions that support quantitative analysis. Our system supports quantitative analysis in two ways. First, it can determine sensitivity by plugging numbers into these symbolic expressions directly. Second, we have implemented a capability to symbolically compute partial derivatives. With this capability, we can determine the relative magnitudes of the partial derivatives with respect to each of the input parameters of the symbolic expression for the measured attribute.

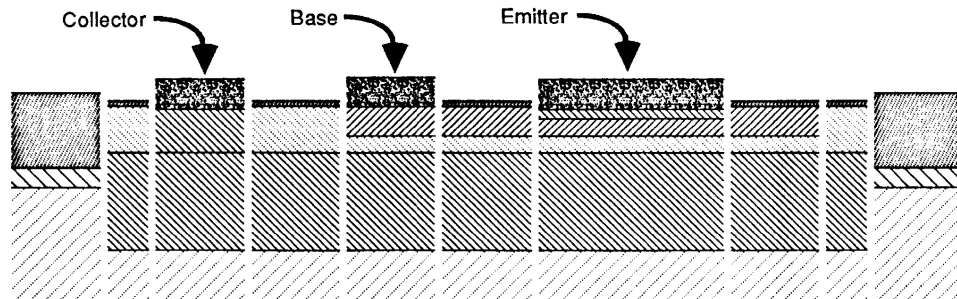


Figure 3: Graphical output of the simulator, showing a NPN bipolar transistor.

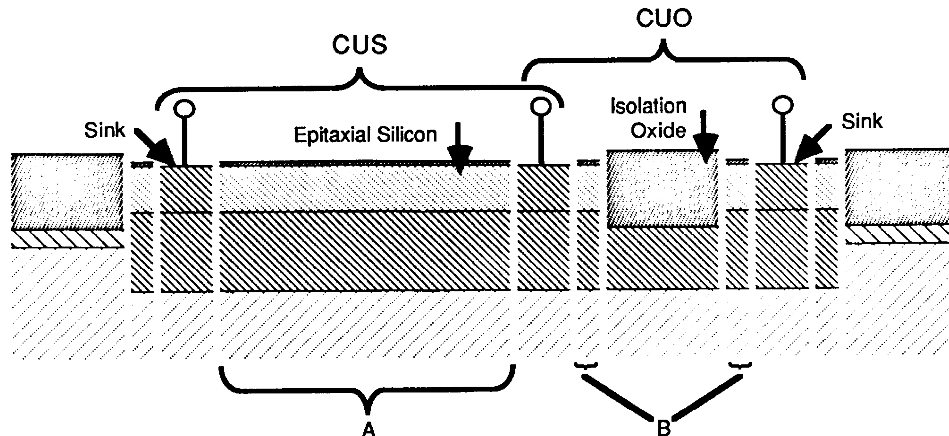


Figure 4: CUS and CUO structures, side by side.

V Future Work

This work suggests several research issues worth investigation.

Currently, we relate the measureable electronic properties of the structures to the geometry of the structures by explicitly giving the system an expression for that dependence. The system then determines the relationship between the geometry of the structures and the input parameters to the processing operations from the wafer history. We are beginning to address the question of how to obtain expressions for the electronic properties of wafer structures by qualitative analysis of the structures using models of electronic behaviour.

The parameter-perturbation fault-model mentioned above implicitly assumes that the problem does not involve gross deviations from the normal structure. If the true problem involves omitting or repeating a processing step, or if the perturbation in the input parameter is very large, then the topology of the wafer structure may be sufficiently modified to make many of the causal pathways in the wafer history inapplicable. In most cases the wafer history for the normal topology will still be a good indicator of which processing steps to suspect. However, it would be advisable to qualitatively simulate the gross errors that are known to occur and "compile" an associative rule-base of causal dependencies from the resulting wafer histories. An expert system (called PIES [Pan]) has already been written that performs the diagnostic task we discuss using associative rules written by production engineers. Currently, each new recipe requires the hand-generation of a new knowledge base—a tedious, time-consuming and error-prone process. The ability to automatically generate a knowledge base for PIES directly from the recipe and the models of the processing operations would greatly enhance its utility.

Finally, we have previously mentioned that qualitative simulation and dependency tracing have a role to play in CAD tools for process designers.

The ability to qualitatively simulate semiconductor manufacturing permits the process designer to take a "top-down" approach to the design of new recipes. The designer can experiment with different sequences of operations, see the results of each sequence and concentrate on obtaining an appropriate sequence, without the necessity of specifying precise numerical values for all the input parameters.

Once the sequence of operations has been chosen and simulated, the causal dependency information can be used to help the designer choose appropriate values for the parameters. First, by determining all the attributes affected by the choice of a value for an input parameter, constraints on the range of values that are appropriate can be determined. As we mentioned earlier, when simulating the recipe qualitatively the designer indicates the desired outcome for each processing step by giving qualitative constraints, such as "the duration of the etch operation is long enough to consume the uppermost layer." These constraints represent design goals to be satisfied by the choice of actual values for the parameters. Second, the expressions for the dependence of attributes on parameters might be used to determine initial estimates for the values of input parameters, by applying constraint propagation and/or numeric root-finding techniques.

VI Conclusion

We have developed qualitative models of the operations performed during semiconductor manufacture. We have represented a recipe for an oxide-isolated bipolar process by a sequence of these models and have simulated the fabrication of several typical devices. The simulation generates a wafer history that describes the complete history of processing for the wafer, from which our system can extract the causal relationships between the attributes that describe the structures on the wafer and the processing operations responsible for creating those structures. Further, the system can determine symbolic expressions for the functional dependence of the these attributes on the parameters to the processing operations. Finally, we have investigated how this information can be used to support a diagnostic reasoning task.

We consider that these models and reasoning processes have an important role to play in computer-aided tools to support many kinds of reasoning tasks in the semiconductor manufacturing domain.

Acknowledgments

The authors thank J. Martin Tenenbaum, Randy Davis and Pat Hayes for reading early drafts of this paper and giving well-considered comments and suggestions.

References

[Fikes and Nilsson] R.E. Fikes and N.J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* **2** (1971) pp189–208.

[Hayes] Pat J. Hayes. The Second Naive Physics Manifesto. In J.R. Hobbs and R.C. Moore (eds.), *Formal Theories of the Commonsense World*, Ablex Publishing Corporation, Norwood, NJ, 1985.

[Ho and Hansen] Charles P. Ho and Stephen E. Hansen. SUPREM III — A Program for Integrated Circuit Process Modeling and Simulation. Technical Report No. SEL 83-001, Integrated Circuits Laboratory, Stanford University, 1983.

[Pan] Y.C. Pan and J.M. Tenenbaum. PIES: An Engineer's "Do it yourself" Knowledge System for Interpretation of Parametric Test Data. In *Proc. AAAI-86*, Philadelphia, PA, August, 1986.

[Sacerdoti] E.D. Sacerdoti. A structure for plans and behavior. Technical Note 109, AI Center, SRI International, Menlo Park, CA, 1975.

[Simmons 83] R.G. Simmons. Representing and Reasoning About Change in Geologic Interpretation. Technical Report 749, MIT AI Laboratory, Cambridge, MA, 1983.

[Simmons 86] R.G. Simmons. Commonsense Arithmetic Reasoning. In *Proc. AAAI-86*, Philadelphia, PA, August, 1986.

[Simmons and Mohammed] R.G. Simmons and J.L. Mohammed. Qualitative Modeling of Semiconductor Fabrication. Technical Report, Schlumberger Palo Alto Research, Palo Alto, CA, in preparation.

[Stefik] M.J. Stefik. An examination of a frame-structured representation system. In *Proc. IJCAI-79*, Tokyo, Japan, August, 1979, pp845–852.