APPLICATION OF NEURAL NETWORK METHOD FOR PROCESS SYNTHESIS

WEN-JUN ZHANG*, JIAN-LIN LIANG, XIAO-FENG XIE, LI-LIN TIAN, ZHI-LIAN YANG

Institute of Microelectronics, Tsinghua University, Beijing 100084, P.R.China
***Email: zwj@mail.tsinghua.edu.cn**

Abstract: Process synthesis is a top-down design methodology and can effectively reduce the process design time. In this paper the general method and the neural network (NN) package used for process synthesis are discussed. Then the characteristics of synthesis some key process modules, includes the ion implantation and well formation, are analyzed, based on which the training set used to build the NN is generated. After that the NN model used for process modules synthesis is constructed and trained. Testing results show that this model can fulfill the process modules synthesis.

1. Introduction

Research on synthetic methodologies in circuit design has caused great changes in the IC design field and shortened the designing cycle sharply. An analogous idea named process synthesis, which meets similar goals in process field, was systematically described [1] recently. This is a top-down methodology in which the known conditions are the desired final process results and the goal is to deduce the correct process steps and parameters.

Inverse modeling of manufacturing process is needed for process synthesis. But normally an inverse model can not be deduced from the process simulating model directly because the forward models based on numerical simulation [2] are usually highly nonlinear. The solution is to collect enough data and build some macro models. And nonlinear modeling technique based on neural networks (NN) has proved to be a powerful tool to build such macro models [3-7]. In this paper a basic inverse model for key process modules is built based on the multi-layer feed-forward NN.

2. NN method for process synthesis

Fig.1 is the general flowchart for the process synthesis by using NN. In the stage of task analysis, two things must be finished. The first thing is to find out the specification of the modeling task (The parameters that can be fixed and those should be set as variables), ignoring the minor factors so that the problem can be more succinct. After that the NN input and output elements should be extracted from the concrete problem. These elements are used instead of the original parameters to build the NN. Such treatment can reduce the network input noise, expand the network synthetic ability and cut down the network scale [3]. Both these two steps are important and based on detail analysis of the modeling requirement. This procedure is similar to the establishment of an "expert system" and needed the experience who builds the model [4].

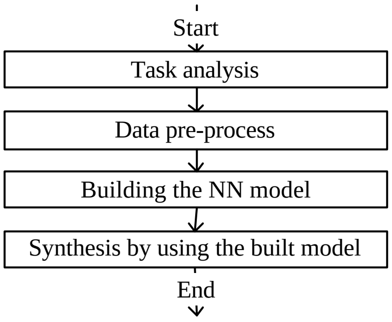

FIG. 1. The flowchart for process synthesis by using NN

The object of data pre-process is to produce the training set of the NN, which represents the relationship of network inputs and outputs. If there is existent data that can be used, the pre-processing work will cost not so much time. Otherwise sufficient experiments are needed to be done to obtain such relationship [5].

After getting the training set, the NN model can be built. In this stage first the network topology structure (The number of network layers and the number of neurons in each layer) is defined, and then the network weight and bias matrices are obtained through training.

When the building stage is finished, the NN model can be used to do the synthetic work. For different process steps and modules, the third and the fourth stages (Building the NN and Synthesis by using the built model) have a lot of common ground. To facilitate their performance, a multi-layer feed-forward NN package that can be used for process synthesis and other purposes is established. This package includes two parts. Part I is written with MATLAB and used for the definition and training of the network. Part II is written with C++. It utilizes the parameter file generated by part I to construct a NN. Once the training work is finished by part I, part Ⅱ

0-7803-7508-4/02/$17.00 ©2002 IEEE

can work as an executive program and can be run without MATLAB. Also, some methods in part II can be called by other C++ programs to build more comprehensive NN synthetic modules. The high effectiveness and reliability of MATLAB makes this package robust enough to establish the purposed NN, while the C++ part makes this package flexible enough to be used for system integration.

3. Process synthesis applications

In the following part we will use the theory and the tool package mentioned above to synthesize the inverse model for ion implantation and N-well formation process.

3.1 Ion implantation

A. Task analysis

This model can deal with implanted profiles with following conditions: one type of substrate, one time of implantation, and using the Gaussian or single Pearson model in simulating procedure.

Our model uses similar structure as in [4], but the inputs are changed to *Rp* and Δ*Rp*, the integral characters of the total profile, and the output is Energy. The Dose parameter is not included in the network output because it can be obtained directly from integrating the profile.

$$Dose = \int_0^{\infty} f(x)dx \tag{1}$$

Here *f(x)* is the concentration at each point.

Using what kind of data to work as the NN inputs is very important for the network scale and capability. In Ref. [4] the input data of the network are point concentrations at 10 fixed coordinates ranging from 0 to 1 µm. This method has some limitations. Normally there is noise at the tail of the profile and such noise can not be eliminated before putting the concentrations into the network. Also, the energy of this method can not be too high. For example, if the energy is 800keV, the occupied space of the profile is from 0 to above 1.5µm. So if with only 10 points that ranging from 0 to 1 µm can not represent the total profile. Of course, this problem can be solved by increasing the number of points, but it will enlarge the scale of the NN.

B. Data pre-process

In order to solve these issues effectively and at the same time to reduce the network scale, we do some pre-process on the profile first, that is, to extract some integral characters of the total profile and use these characters instead of point concentrations to build the network. When the profiles are generated by using Gaussian or Pearson model, *Rp* and Δ*Rp* are such characters.

$$Rp = (\int_{-\infty}^{\infty} xf(x)dx)/(\int_{-\infty}^{\infty} f(x)dx) \tag{2}$$

$$\Delta Rp = \sqrt{(\int_{-\infty}^{\infty}(x - Rp)^2 f(x)dx)/(\int_{-\infty}^{\infty} f(x)dx)} \tag{3}$$

Normally *f(x)* is 0 when *x*<0 (outside the substrate). Equations (2) and (3) can be changed into

$$Rp = (\int_{0}^{\infty} xf(x)dx)/(\int_{0}^{\infty} f(x)dx) \tag{4}$$

$$\Delta Rp = \sqrt{(\int_{0}^{\infty}(x - Rp)^2 f(x)dx)/(\int_{0}^{\infty} f(x)dx)} \tag{5}$$

Since the relationship between *Rp*, Δ*Rp* and Energy has been made in the lookup tables of simulators such as Tsuprem-4 [2]. We can use these data in the lookup table to construct the NN model.

C. Building the neural network model

It has been proven that multilayer feedforward NN with a nonpolynomial activation function can approximate any continuous function to any degree of accuracy [6]. Combining with the experience in [4], we construct a NN with two hidden layers and one output layer. The neurons in these layers are 7, 5, 1 (as in Fig.2 and Fig. 3), and the transfer functions are log sigmoid, log sigmoid, linear, respectively. The training set is part of the lookup table for boron with channeling in silicon [2], covering Energy from 5keV to 850keV.

The training is done by MATLAB. The algorithm is Levenberg-Marquardt back-propagation. The epochs are 6000 and the mean square error (MSE) is 0.006.

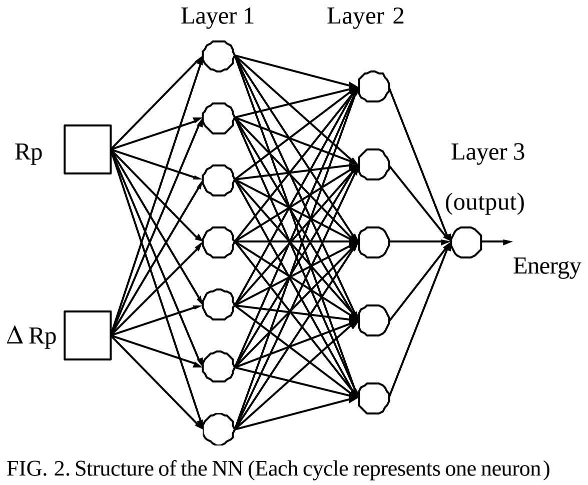

FIG. 2. Structure of the NN (Each cycle represents one neuron)

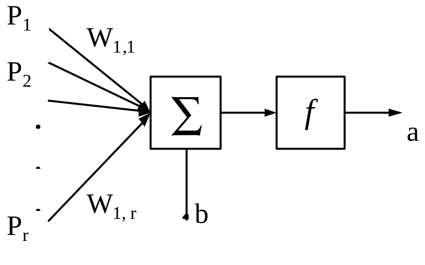

FIG. 3. Structure of one single neuron [7]: *a = f(Wp+b)*

C. Synthesis by using the built model

Table 1 shows the result of this model. It can be seen that the synthetic result matches the simulating Dose and Energy quite well.

Comparing with the model in [4], this basic model has a less network scale (2 inputs instead of 10 inputs) but can fit for a wider range of Energy (for boron, the range of Energy is expanded to 5keV–800keV). In addition, this basic model can effectively eliminate noise at some points since its inputs are integral characters of the total profile.

TABLE 1. Test result of the basic model

(Boron, Pearson model, simulating Dose=2.0e13 cm-2)

Simulating Energy (keV)	Simulating Dose (* 1e13cm-2)	Synthesis Energy (keV)	Synthesis Dose (*1e13cm-2)
5	2.0	5.0312	1.9939
15	2.0	15.0056	1.9995
33	2.0	33.0799	2.0001
55	2.0	54.9713	2.0002
84	2.0	83.7428	2.0002
550	2.0	545.656	2.0003

3.2 N-Well formation

A. Task analysis

Well formation is normally the first and essential module of the MOS process.

The purposed final wafer state of this module, which should be included in the NN inputs, is some of well characteristics. One of them is the surface concentration of silicon layer. The reason is that it can have substantial influence on the threshold voltage thus have effect on the process module of threshold voltage adjustment. Another well characteristic is well depth.

Now let's analyze the process steps and parameters of N well generation. It usually includes 3 key steps: (1) Initial pad oxidization; (2) Ion implantation; (3) Drive-in (with high temperature and long time)

The purpose of the first step is to generate an amorphous layer on the silicon surface. The thickness of the SiO_2 is about 200 Å ~ 400 Å. This SiO_2 layer will be etched immediately after the well formation.

Step 2 provides an original total dose for the drive-in. Normally the implanting energy is set to a constant, while the implanting dose can be changed to obtain different final surface concentration of silicon layer.

The purpose of step 3 is to obtain the relatively flat impurity distribution near the silicon surface and get the needed well depth. The temperature is very high (Normally it is no less than 1000 °C) and the drive-in time is long (normally it is no less than 2 hours).

Two types of gas can be used during the step: oxygen and nitrogen. We select nitrogen here for the goal of simplifying the issue. Under such condition the thickness of the oxide layer will not be changed during the drive-in step. So when building the synthetic model the thickness of this layer can be fixed, which here is 250Å.

Based on the above analysis, the number of process variables is reduced from 5 (the thickness of the initial pad oxide, implanting dose and energy, drive-in temperature and time) to 3 (implanting dose, drive-in temperature and time). Others are set to constants. But not all the 3 variables should be used as NN outputs. The reason is that drive-in is a multi-to-one process. That means different sets of drive-in temperature and time may obtain the same curve shape of impurity. For a given impurity curve we can not determine its drive-in parameters unless one of them is known. So we assume that the drive-in temperature is known and set it as the third NN input. And the implanting dose and drive-in time are set as outputs.

To ensure that the training work of the NN can be carried out, some mathematical process should be done on the original parameters. The actually NN inputs we use are: drive temperature, well depth, and base 10 logarithm of the silicon surface concentration minus 16. And the network outputs are: base 10 logarithm of the implanting dose minus 12, one tenth of the drive-in time.

B. Data pre-process

For the deep sub-micron process, the typical surface concentration of the silicon layer is about 5e16 cm^{-2}, and the typical well depth is around 2μ m. So the training set to be used, which represents the relationship of the network inputs and outputs, should cover this range. If we use the substrate concentration of 6e14cm^{-2}, the well depth is defined as the distance from the point whose impurity concentration equals to 6e14 cm^{-2} to the surface point.

The conditions we use to generate the training set are:

- Substrate concentration: 6e14cm^{-3}
- Thickness of the initial pad oxide layer: 250 Å
- Implanting energy: fixed at 100 keV
- Implanting dose (cm^{-2}): 3.9e12, 4.3e12, 4.4e12, 4.5e12, 4.7e12, 5.1e12
- Gas used during in drive-in: nitrogen
- Drive-in temperature(°C): 1050, 1070, 1090
- Drive-in time(hours): 2, 4, 6, 8, 10, 12, 14, 16, 18

Under these conditions 162 curves are simulated. The network inputs and outputs are extracted from each curve to compose the training set. In this training set the well depth ranges from 0.93 μ m to 3.65 μ m while the surface concentration changes from about 1.23e17 to 2.66e16cm^{-3}.

Fig. 4 is a 2D figure and it gives out the actual range of well depth and surface concentration combination for the network inputs when the drive-in temperature equals 1050

°C. And Fig.5 is a 3D figure that gives out the actual range of network inputs for the drive-in temperature ranges from 1050 °C to 1090 °C.

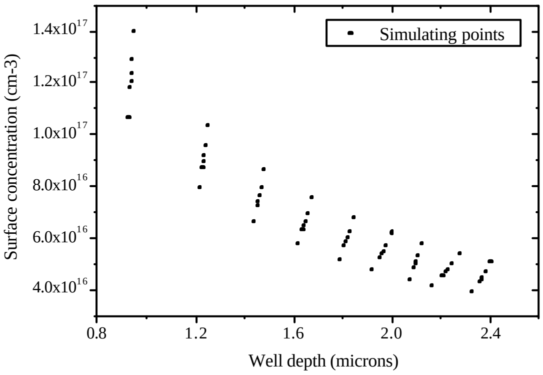

FIG. 4. The range of the network inputs when drive-in temperature is 1050 C

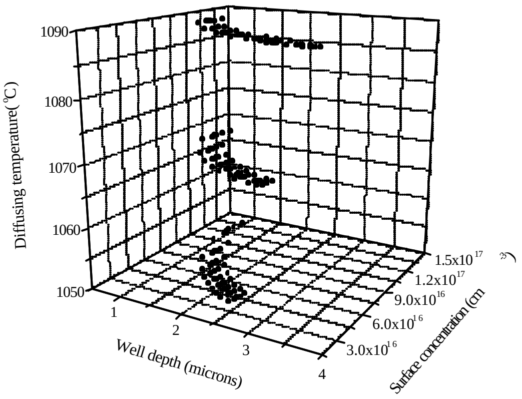

FIG. 5. The actual range of network inputs for the drive-in temperature ranges from 1050 °C to 1090 °C

C. Building the neural network model

The network structure is defined as:

- 3 inputs: Drive-in temperature; Well depth; Logarithm of surface concentration minus 16;
- 2 outputs: Logarithm of the Dose minus 12; One tenth of the drive-in time;
- 4 layers of neurons;
- The number of neurons in each layer is 18, 12, 9, 2.

After the definition, the network is trained by using the general NN package. One thing should be mentioned is that all the training work is done under MATLAB environment. The training epochs are 1400 and the final mean square error (MSE) is 1.44e-4. The algorithm used in the first 200 training epochs is RPROP back-propagation algorithm. And in the rest epochs it is BFGS quasi-Newton back-propagation algorithm.

C. Synthesis by using the built model

All the points (Not only the points used for training) in the regions shown in Fig. 4 (drive-in temperature =1050 °C), or in Fig. 5 (drive-in temperature ranges from 1050 °C to 1090 °C) can be used as the synthetic targets.

It is shown that the built model can complete the synthesis of well formation (Table.2). When drive-in temperature equals 1060°C, the result of the example in table 1 is not as exact as others. The reason is that for this example the target needed to be synthesized is slightly out of the training range (Refer to Fig.5).

TABLE 2. Testing result of the well formation synthesis

Purposed well depth (μ m)	2	
Purposed surface concentration (cm^{-3})	5e16	
Purposed drive-in temperature (°C)	1050	1060
Synthetic implanting dose (cm^{-2})	4.2160e12	3.9564e12
Synthetic drive-in time (hours)	12.743	9.422
Actual well depth (μ m)	2.010	1.927
Actual surface concentration (cm^{-3})	4.949e16	4.786e16
Relative error of well depth	0.5%	3.65%
Relative error of surface concentration	1.02%	4.28%

4. Conclusion

Process synthesis is a newly important research field due to its ability to cut down the process designing time greatly. In this field the given conditions are the final wafer state and the object is to find out the appropriate process steps and parameters that can obtain such results. The nonlinear modeling methods based on NN have proved to be the powerful tools.

In this paper, we introduce the theory and the tool package for process synthesis by using NN. As examples we use them to achieve the synthesis of ion implantation and well region formation. Testing results show that the synthetic model we build works effectively.

References

[1] Hosack H H, Mozumder P K, Pollack G P. Recent advances in process synthesis for semiconductor devices. IEEE Trans. on Electronics Devices, 1998, 45(3): 626-633.

[2] SUPREM4 user's manual. TMA Inc., 1995.

[3] May G. Manufacturing ICs the neural way. IEEE Spectrum, 1993, 31(9): 47-51.

[4] Pantic D, Trajkovic T, Stojadinovic N. A new technology computer-aided design (TCAD) system based on neural network models. Microelectronics Journal, 1998, 29: 1-4.

[5] Nadi F, Agogino A M, Hodges D A. Use of influence diagrams and neural networks in modeling semiconductor manufacturing processes. IEEE Trans. on Semiconductor Manufacturing, 1991, 4(1): 52-58.

[6] Leshno M, Lin V Y, Pinkus A, Schocken S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Net works, 1993, 6: 861-867.

[7] Neural Network Toolbox User's Guide. The MathWorks, Inc., 1999.