INVITED PAPER

A Neural Network-Based Prediction Model in Embedded Processes of Gold Wire Bonding Structure for Stacked Die Package

This modeling technique uses artificial neural networks to make the best processing choices to achieve high-quality bonding of the wires used to interconnect stacked components.

By Chin-Huang Chang and Yung-Hsiang Hung

ABSTRACT | The trend in the consumer electronics market is to offer lighter, smaller outline, and more functional products, especially for portable products. This trend has pushed electronic packages to be thinner, with a lower profile and with multiple chips in one package. When the package becomes thinner, the space of the package correspondingly becomes an important issue. In order to obtain higher density and thinner package, we have to develop an embedded gold wire bonding assembly technology. It provides a simple structure with a thick adhesive layer to fix the upper layer die and bottom layer die gold wire. The developed technology can use exactly the same die size as for wire bonding interconnections without any additional processing. All electrical connections of the upper and the lower die are achieved by wire bonding to the substrate, independently. We have performed this stacking assembly by precise control of the die attach film layer thickness and low wire loop shape.

KEYWORDS | Artificial neural networks; dicing die attach film; die stacking; embedded process; gold wire bonding; same size

Manuscript received September 29, 2007; revised December 2, 2007. Current version published February 27, 2009. This work was supported by the National Science Council, R.O.C., under Contract NSC-96-2221-E-167-022.

C.-H. Chang is with the R&D Center, Siliconware Precision Industries Co., Ltd., Taichung, Taiwan, R.O.C. (e-mail: vic@spil.com.tw).

Y.-H. Hung is with the Department of Industrial Engineering and Management, National Chin-Yi University of Technology, Taichung 411, Taiwan, R.O.C. (e-mail: hys502@ncut.edu.tw).

Digital Object Identifier: 10.1109/JPROC.2008.2007468

I. INTRODUCTION

In recent years, cellular phones, personal digital assistants, and other portable devices have grown rapidly in popularity, thanks to compact, thin, and light features that have become key requirements for design. As packaging sizes continue to decrease, the level of integration of semiconductor devices continues to increase both in complexity and in number of components. Today, IC assemblers are well suited to developing new packages. In fact, most semiconductor IC packaging houses have developed fine-pitch ball grid array packages using dummy silicon or polyimide interposers. Dozens of new package styles have been developed for specific applications.

In IC packaging patterns, gold wire package is one of the main packaging technologies at present, handling about 90% of the total demand. The wire bonding process is mainly used to transmit signals among components. For the IC package products with a high unit price, the quality of the wire bonding process has the greatest effect on the postprocess and the yield loss of products, handling about 40%. In recent years, portable electronic communication products have become increasingly thinner, which makes the IC chip package body thinner as well. Meanwhile, demand for the stacked die package, such as chip scale package (CSP) and some micro package technology, has gradually matured. Thus, in designing new products, it is necessary to develop the processing technology and the production flow for these new products in order to satisfy the demands of customers' design specifications.

0018-9219/$25.00 ©2009 IEEE

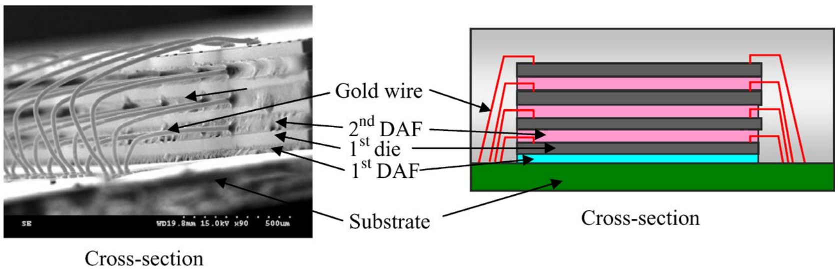


Fig. 1. ***Embedded gold wire bonding for four die stacked package.***

Embedded gold wire technology was developed in order to reduce the Z (vertical) dimension of the IC package. The process flow of embedded gold wire bonding for die-to-die stacking places the first die to substrate using thinner die attach film. The die attach film was laminated onto the backside of the wafer followed by the die saw and die attach. It is called the first die attach process. After the first die attach, the product is sent to the wire bond station for the first wire bond. In the wire bond process, the key dimension is to control the gold wire loop height within 45 μm. Because the gold wire loop height is the key quality characteristic for the embedding process, it will affect the clearance between die-to-die bonding Z dimension. After the first wire bond with low gold wire loop, the following process is the second die attach. Since the second die attach process is die-to-die attach, it is laminated with thicker die attach film onto the backside of the second wafer. The second die attach film should cover the first gold wire on the first die surface. The key quality characteristic of the second die attach process is die attach film void. This kind of process will be repeated four times for the four die stacked package.

There are some key parameters of embedded gold wire bonding assembly technology for the same die size stacking package: for example, film type, bond force, bond time, bond temperature, and curing temperature. The key challenges of the embedded gold wire process for the same die size die attach are how to control the loop height during the die attach and to control the die attach film void during the die attach. Since there are four times of die attach and four times of wire bond, different die attach parameters will significantly impact the quality of this product.

This paper introduces the method for determining the key parameters of the process to ensure the robustness of the design of the embedded gold wire bonding structure for stacked die package. Fig. 1 shows the structure of embedded gold wire bonding for the four die stacked package.

II. LITERATURE REVIEW

In the processes of research and development of IC products, widely used statistical research methods are experimental of design (DOE) [1], response surface methodology (RSM) [2], finite element method (FEM) [3], [4], and Taguchi method [5]. The advantage of using the Taguchi method lies in the reduction of both production cost and time. It concerns minimizing the effect of uncertainty or variation in design parameters. However, the execution time for these types of analyses can be on the order of hours or days for the experimental evaluation. Indeed, most Taguchi procedures have been applied to analyzing only linear systems, under the assumption of the addition of individual factor effects [6]. In recent years, artificial neural networks (ANNs) have been used in a large number of applications and have proven to be effective in performing complex functions in various fields. ANNs can be structured to perform classification [7], approximate equations [8], and predict values [9], and they have successfully solved problems through generalization of a limited quantity of training data, concerning overall trends in functional relationships [10], [11]. The widely used algorithms of ANNs in approximate models are the back-propagation neural networks (BPNNs) and radial basis function networks (RBFNs). The most successful BPNN learning model until now is the least mean square algorithm, due to its high prediction accuracy's having been excellently proven. However, despite the practical success, BPNNs suffered from increasing convergence since building an optimal prediction model with multiple quality characteristics is complicated by the presence of many training factors. Training factors typically involved in building a neural network model may include: the hidden neuron, training tolerance, initial weight distribution, and function gradient. The most difficult problem often arises from the nature of randomness in the initial weight distribution [12]. The other widely used algorithms of ANNs in approximate models are the RBFNs. The RBFNs use exponentially decaying localized nonlinearities to construct nonlinear approximations in a higher dimensional space. It is a function that depends only on the radial distance from a point and a special linear layer. The RBF network is a popular alternative to the NNs, which can offer advantages over the BPNN in some applications. In fact, an RBF network can be easier to train than a BPNN network. The RBF network has a similar form to that of

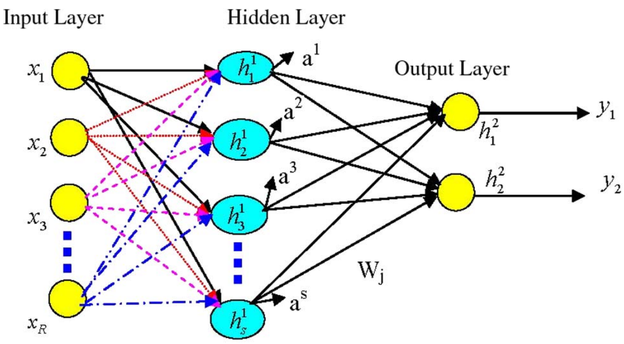


Fig. 2. ***The architecture of the RBFN.***

the multilayer perceptron (MLP): a multilayer, forward network. However, unlike the MLP, the hidden units in the RBF are different from the units in the input and output layers: they contain the "radial basis function," a statistical transformation based on a Gaussian distribution from which the neural network's name is derived. Like MLP neural networks, RBF networks are suited to applications such as pattern discrimination and classification, pattern recognition, interpolation, prediction and forecasting, and process modeling. Training an RBF net is very quick because the nonlinear part, the width and center points, is solved by setting the parameters to some (more or less) predefined values. The remaining problem is linear and consequently quickly solved.

III. THE RBFN CHARACTERISTICS

The design of an RBFN in its most basic form consists of three separate layers: input layer, hidden radial basis layer of H^1 neurons, and output linear layer of (H^2) neurons. The hidden layer contains a number of RBF neurons $(h_1^1, h_2^1, \ldots, h_s^1)$, and each of them represents a single radial basis function. The output layer with linear activation functions is (h_1^2, h_2^2). Fig. 2 shows the architecture of an RBFN.

Where X is the input vector, R is the number of elements in input vector. As such, the input layer is extended to become: $x_1, x_2, \ldots, x_R$. In general, radial basis neurons with weight vectors quite different from the input vector X have outputs near zero. There are many definitions of the nonlinear function. In the RBFN model, in particular, the output of the first layer for a feed-forward network net can be obtained with the nonlinear activation function as follows:

$$\begin{aligned} a^1 &= \Phi_N(x) = \Phi|x - c_N| \\ &= \exp\left\{-\sum_{j=1}^{N}\left[\frac{(x_j - c_{ij})^2}{2\sigma_{ij}^2}\right]\right\}, \quad i=1,2,\ldots,n. \end{aligned} \tag{1}$$

Where a^i is the output of the nonlinear transformation of the neuron i in the hidden layer, w_j is the weight of the output layer, the C_{ij} and σ_{ij} are decoupled with the determination of w_j, and output $Y = \sum w_i \Phi_i(x)$. Each neuron's weighted input is the distance between the input vector and its weight vector, calculated with distance. However, these small outputs have only a negligible effect on the linear output neurons. In contrast, a radial basis neuron with a weight vector close to the input vector X produces a value near one. If a neuron has an output of one, its output weights in the second layer pass their values to the linear neurons in the second layer. Therefore, if only one radial basis neuron had an output of one, and all others had outputs of zeros (or very close to zero), the output of the linear layer would be the active neuron's output weights. In short, if a neuron's weight vector is a distance of spread from the input vector, its weighted input is spread: its net input is $\sqrt{-\log(0.5)}$ (or 0.8326); therefore, its output is 0.5. The bias b_i of the hidden layer neurons allows the sensitivity of the neuron to be adjusted; this is determined by

$$b_i = \frac{\sqrt{-\log(0.5)}}{\sigma_i}. \tag{2}$$

As stated above, during the designing of the RBFN, it should be ensured that b_i is large enough so that the active input regions of the radial basis neurons sufficiently overlap, so that several radial basis neurons always have fairly large outputs at any given moment. In fact, b_i should not be so large that each neuron is effectively responding in the same large area of the input space. Therefore, the value of b_i greatly affects the performance of RBFN. In the learning strategies, a supervised learning algorithm determined the centers (σ), widths (b), and weights (w) of the output layer. Gradient descent algorithm can be used to minimize the error between the actual and the desired output over the training set. The error was measured by the typical root mean squared error (RMSE) defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{p=1}^{n}\sum_{j=1}^{s}\left(T_j^p - Y_j^p\right)}{m \times n}} \tag{3}$$

where n is the number of the training sample, s is the number of neurons, and T_j^p and Y_j^p represent the desired and calculated outputs of the jth output neuron for the pth training sample. Besides, the training goal (RMSE goal) also can affect the performance of the RBFN model. Too small a goal value causes overtraining, while too big a goal value influences the accuracy of the RBFN model, so the training goal value has been optimized.

Table 1 Experimental Factor and Results

Key Item	Control Factor					Quality Characteristics	
	Film type	Bond force (N)	Bond time (Sec.)	Bond temp. (Deg'C)	Curing temp. (Deg'C)	Loop ht.(um) 45+/-10 (NTB)	Film void (%) 15% max. (STB)
Variable Name	A	B	C	D	E	Y1	Y2
Experiment NO. 1	H	8	3	130	150	46	5
2	H	8	3	130	150	47	4
3	H	8	3	130	150	45	4
4	H	8	3	130	130	46	6
5	H	8	3	130	130	45	4
6	H	12	3	130	130	42	0
7	H	12	3	130	130	41	0
8	H	12	3	130	130	40	0
9	H	12	3	130	130	40	0
10	H	12	3	130	130	41	0
11	M	8	3	120	150	45	89
12	M	8	3	120	150	45	86
13	M	8	3	120	130	46	8
14	M	8	3	120	130	48	10
15	M	8	3	120	130	45	6
16	M	8	2	130	150	47	85
17	M	8	2	130	150	44	84
18	M	8	2	130	150	46	82
19	M	8	2	130	130	46	8
20	M	8	2	130	130	45	12
21	V	8	2	120	150	48	0
22	V	8	2	120	150	50	0
23	V	10	2	120	150	44	0
24	V	10	2	120	150	43	0
25	V	10	2	120	150	44	0
26	V	8	2	130	130	46	0
27	V	8	2	130	130	49	0
28	V	10	2	130	130	43	0
29	V	10	2	130	130	42	0
30	V	10	2	130	130	40	0

IV. EXPERIMENT AND RESULT ANALYSIS

This section deals with building the experimental factors of embedded gold wire bonding of stacked die package through the experiments' method and constructing two important quality characteristics. Next, the levels of experimental factors are normalized as the prediction variables for a wire bonding RBF neural network (WB_RBFNN) and the loop height and film void from experiments are set as the output of WB_RBFNN to construct the nonlinear network models. The five control factors included in 30 trail experiments are film type, bond force, bond time, bond temperature, and curing temperature for the embedded gold wire bonding of a stacked die package. From the experiment result, there are a total of 30 observations in our study. The data set contained five

Table 2 Design Parameters for BPNN and WB_RBFNN

BPNN Model		
training function		trainlm
activation function	hidden layer	login
	output layer	purelin
net.trainParam.lr		0.4
net.trainParam.mu		0.1
net.trainParam.mu_dec		0.1
net.trainParam.mu_inc		10
net.trainParam.mc		0.5
net.trainParam.epochs		1000
net.trainParam.goal		0.001

WB_RBFNN Model	
Mean squared error goal	0.0
Spread of radial basis function	0.04
Maximum number of neuron	30
frequency of progress display	10

Table 3 RBFN and BPNN Network Simulation Results

Trail	RBFNs' RMSE			BPNNs' RMSE		
	Training	Test	Error_fit	Training	Test	Error_fit
1	0.9991	0.9270	0.0721	0.9700	1.2162	0.2462
2	1.0138	0.9895	0.0243	0.9878	1.1180	0.1302
3	1.0017	1.0206	0.0189	0.9913	1.0992	0.1079
4	1.0104	0.9789	0.0315	0.9929	1.2458	0.2529
5	1.0069	0.9574	0.0495	0.9939	1.0533	0.0594
6	1.0017	0.9813	0.0204	**0.9014**	**1.3385**	**0.9652**
7	**0.9957**	**1.0026**	**0.0069**	1.0052	0.9789	0.0263
8	0.9939	0.9175	0.0764	0.9691	1.2054	0.2363
average	1.0029	0.97185	**0.0375**	0.97645	1.156913	**0.25305**
Mean Time (s)	0.016917			0.3862		
Min. Time (s)	0.00495			0.12475		
Max. Time (s)	0.0109867			0.293521		
Std. (s)	0.004109528			0.092875048		

input variables (control factors) and two target variables. Whichever, there are two kinds of input variable in the data set: four continuous variables (factors B, C, D, and E) and one nominal variable (factor A). Before training, it is often useful to scale the input variables and targets so that they always fall within a specified range. In the study reported in this paper, the Table 1 data were preprocessed with a transformation encoding; one binary coding scheme was applied to the nominal variables. For example, the "film type" (factor A) variable was encoded as: a binary code "level H" $\rightarrow (-1, -1, -1)$, "level M" $\rightarrow (0, 0, 0)$, and "level V" $\rightarrow (+1, +1, +1)$ in Table 1. Next, the training data were randomly selected from the data set, and set as the output of WB_RBFNN to construct the nonlinear network models. Here, 80% input patterns (24 observations) are used for training and the remaining 20% observations (six observations) are used for testing.

The WB_RBFNN simulation was performed using MATLAB software. The system configuration of the computer used for training both models was as follows: operating system Windows XP; processor Intel Duo T7500; and total physical memory 1.5 GB. The main design parameters of the BPNN and WB_RBFNN models are shown as Table 2.

Table 4 WB_RBFNN Experiment Results Comparison

Quality Characteristic		Training									
		1	2	3	4	5	6	7	8	9	10
Loop height	Experit.	45	45	44	48	45	40	46	45	45	41
	WB_RBFNN	45.5	46.333	45.457	46.134	46	41.5	45.5	45	45	41
Film void	Experit.	4	6	84	10	4	0	8	86	89	0
	WB_RBFNN	5	8	83.667	8	4	0	10	87.5	88.5	0
Quality Characteristic		Training									
		11	12	13	14	15	16	17	18	19	20
Loop height	Experit.	46	49	44	42	47	45	40	47	43	46
	WB_RBFNN	45.5	49	43.5	41	46	45.5	41	45.667	43.5	45.667
Film void	Experit.	6	0	0	0	4	12	0	85	0	82
	WB_RBFNN	5	0	0	0	4	10	0	84.167	0	82.667
Quality Characteristic		Training				Testing					
		21	22	23	24	25	26	27	28	29	30
Loop height	Experit.	48	43	50	46	41	46	42	44	46	40
	WB_RBFNN	49	41.5	49	46.133	41	49	41.5	43.5	46	41
Film void	Experit.	0	0	0	8	0	0	0	0	5	0
	WB_RBFNN	0	0	0	8	0	0	0	0	4	0

From Table 3, RBFNs' Error_fit average is 0.0375 and better than BPNNs' 0.25305. In the BPNN, it was observed that sixth trail was overfitting. Generally speaking, RBFN performed better than BPNN, so this study uses RBFNs' seventh trail as the WB_RBFNN forecast model. Table 4 shows the comparison for conform experiment results and WB_RBFNN prediction results.

V. CONCLUSION

This paper applies ANNs to discover the production combination of production levels for control factors of embedded gold wire bonding assembly technology for the same die size stacking package. Based on this ANN model, it becomes easy to forecast the quality output by using different kinds of parameter combinations. This model also can be applied to the design stage of new products to foresee the design quality. ■

Acknowledgment

The authors would like to thank the anonymous referees for their careful reading of this paper and several suggestions that improved this paper.

REFERENCES

[1] R. Groover, W. K. Shu, and S. S. Lee, "Wire bond loop profile development for find pitch-long wire assembly," *IEEE Trans. Semicond. Manuf.*, vol. 7, pp. 393–399, 1994.

[2] B. Shu, "Wire bond development for high-pincount surface-mount," in *Proc. 42nd Electron. Compon. Technol. Conf.*, 1992, pp. 890–898.

[3] D. S. Liu, Y. C. Chao, and C. H. Wang, "Study of wire bonding looping formation in the electronic packaging process using the three-dimensional finite element method," *Finite Elements Anal. Design*, vol. 40, pp. 263–286, 2004.

[4] I. Chaudhry and F. Barez, "Optimization of wire loop height for a cavity down plastic pin gird array package," *Trans. ASME J. Electron. Packag.*, vol. 113, pp. 194–200, 1998.

[5] H. C. Chen, "Study on wire bonding process for chip scale packaging," master's thesis, Dept. of Industrial Mechanical Engineering, National Cheng Kung University, Taiwan, R.O.C., 2000.

[6] L. I. Tong, C. T. Su, and C. H. Wang, "The optimization of multi-response problems in Taguchi method," *Int. J. Qual. Reliab. Manage.*, vol. 14, pp. 367–380, 1997.

[7] I. M. Raimundo and R. Narayanaswamy, "Simultaneous determination of relative humidity and ammonia in air employing an optical fibre sensor and artificial neural network," *Sens. Actuators B, Chem.*, vol. 74, no. 1–3, pp. 60–68, 2001.

[8] B. S. Joo, N. J. Choi, Y. S. Lee, J. W. Lim, B. H. Kang, and D. D. Lee, "Pattern recognition of gas sensor array using characteristics of impedance," *Sens. Actuators B, Chem.*, vol. 77, no. 1/2, pp. 209–214, 2001.

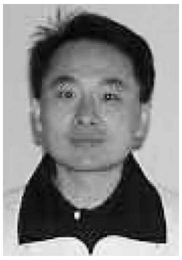
[9] J. Freeman and D. Skapura, *Neural Networks, Algorithms, Applications, and Programming Techniques*. Reading, MA: Addison-Wesley, 1991, pp. 89–111.

[10] N. Sandro, "Feedforward neural networks for principal components extraction," *Comput. Statist. Data Anal.*, vol. 33, no. 4, pp. 425–437, 2000.

[11] R. J. Schalkoff, *Artificial Neural Networks*. New York: McGraw-Hill, 1997.

[12] B. Kim, D. W. Kim, and G. T. Park, "Prediction of plasma-induced DC bias using polynomial neural network," *Vacuum*, vol. 79, pp. 111–118, 2005.

ABOUT THE AUTHORS



Chin-Huang Chang received the B.S. degree from Yunlin Institute of Technology, R.O.C., and the master's degree in industrial engineering from Feng Chia University, Taichung, Taiwan, R.O.C.

He was previously an R&D Engineer with system in package IC products and assembly process in IC packaging of the semiconductor industry. He is currently a Senior Manager with the R&D Center, Siliconware Precision Industries Co., Ltd. His research interests include artificial intelligence, management of technology and innovation, system in package IC products, and process development.

Yung-Hsiang Hung received the Ph.D. degree in industrial engineering and management from National Chiao-Tung University, R.O.C., in 2002.

He is an Associate Professor of industrial engineering and management at National Chin-Yi University of Technology, Taiwan, R.O.C. His main research interest is in the area of statistical process control and service quality management. His research areas include process capability analysis, semiconductor manufacturing management, and process parameter optimization design.