A Probabilistic Neural Network Hardware System Using A Learning-Parameter Parallel Architecture

Noriyuki Aibe*, Moritoshi Yasunaga†, Ikuo Yoshihara††, and Jung H.Kim†††

*Master Program of Information Sciences and Electronics, University of Tsukuba, Tsukuba, Ibaraki. 305-8573 Japan.
†Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba, Ibaraki. 305-8573 Japan.
††Faculty of Engineering, University of Miyazaki, Miyazaki, Miyazaki. 889-2192 Japan.
†††System Engineering Dept., University of Arkansas at Little Rock, Little Rock, AK 72204-1099, USA.

E-mail: *susu@susutawari.org, †yasunaga@is.tsukuba.ac.jp, ††yoshiha@cs.miyazaki-u.ac.jp, †††jhkim@ualr.edu

Abstract - A novel PNN (Probabilistic Neural Networks) hardware architecture called 'Sigma Parallel Architecture'(SPA) is proposed. Different values of the network parameter are calculated in parallel in the SPA and it speeds up the PNN learning as well as recognition overcoming the difficulty in the VLSI implementation. The hardware prototype is developed using FPGA chips and it shows a high speed leaning of about 10 seconds that satisfies the requirements in the real world image recognition tasks.

I. INTRODUCTION

The Probabilistic Neural Network(PNN), which is a three-layer feed-forward network with a single learning parameter proposed by Specht[1][2], is one of the promising algorithms for the real world pattern recognition applications. Because PNN is constructed based on the Bayes' theorem, high recognition accuracy may be obtained with a large number of sample patterns. Recently, real world pattern recognition problems such as a satellite cloud classification[3][4], facial image recognition[5], etc. have been applied to PNN. The major deficiency of PNN, however, is that it requires long computation time even in the recognition (i.e., an one-shot calculation in the network) because of the large number of sample patterns that are mapped onto each neuron.

Consequently, it is difficult to determine the network parameter (smoothing parameter σ) in real time since the recognition calculation needs to be repeated changing the parameter. The parameter σ can be calculated theoretically if the distribution of sample patterns can be derived from the theory. However, for the real world pattern-recognition tasks, it is impossible to assume the distribution theoretically in general. Thus, searching the best σ in the large search-space is indispensable in PNNs applied to the real world tasks (genetic algorithms are tried to solve this problem[5]). Minchin et al. proposed a hardware design of PNN[6]. Their pioneer work showed that the design was effective to speed up the recognition process in PNN with the simple hardware architecture. However, no learning circuit or architecture has been proposed yet.

In this paper, we propose a high speed learning architecture 'Sigma-Parallel Architecture'(SPA) in order to overcome the difficulty that occurs to the parallel implementation of the PNN learning onto the VLSI. We develop the prototype system of PNN with the proposed learning architecture, and demonstrate its efficiency comparing it with the 1GHz PentiumIII PC through simulation.

II. PROBABILISTIC NEURAL NETWORKS

PNN consists of feed-forward three layers of neurons, as shown in Fig. 1. Each neuron in the kernel layer corresponds to the kernel function , where $\mathbf{X}$ is the input pattern and $\mathbf{S}_j^i$ is the jth pattern in the category C_i. The superposition of the kernel functions that is executed in the summation layer, constructs the estimator of the probabilistic density function $p\left(\mathbf{X}|C_i\right)$ of the category C_i,

Authorized licensed use limited to: Universita degli Studi di Salerno. Downloaded on May 25,2023 at 07:56:10 UTC from IEEE Xplore. Restrictions apply.

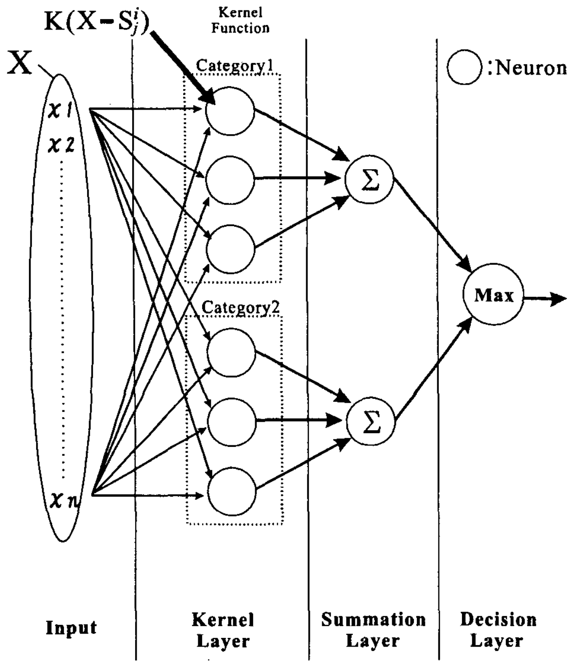

Fig. 1. Network Configuration of PNN.

that is,

$$\hat{p}\left(\mathbf{X}|C_i\right) = \frac{1}{N_p}\sum_{j=0}^{N_p} K\left(\mathbf{X}-\mathbf{S}_j^i\right), \tag{1}$$

where N_p is the number of sample patterns in each category and N_p is regarded as 1 because it can be assumed to be identical in all categories. The neuron in the decision layer selects the highest and its category C_i is output as the answer (i.e., predicted category) to the input pattern (unknown pattern) $\mathbf{X}$.

The Gaussian basis function

$$K\left(\mathbf{X}-\mathbf{S}_j^i\right) = \exp\left(\frac{-(\mathbf{X}-\mathbf{S}_j^i)^T(\mathbf{X}-\mathbf{S}_j^i)}{2\sigma^2}\right), \tag{2}$$

or the Parzen window function (uniform function)

$$K\left(\mathbf{X}-\mathbf{S}_j^i\right) = \begin{cases} 1: & \left|x_k-(s_j^i)_k\right| \le \frac{\sigma}{2} \\ 0: & \text{otherwise} \end{cases} \tag{3}$$
$$(\text{for all } k, k=1,2,...,n)$$

is usually used as the kernel function. The smoothing parameter σ determines the width of the kernel function. Determining σ, which is just the learning in PNN, seems to be easy because it is the only one parameter in the network. However, the estimator $\hat{p}\left(\mathbf{X}|C_i\right)$ responds so sensitively to σ that high accuracy is required to determine σ. Furthermore, there is no general method to determine σ, so that σ is usually determined by changing its value minutely and examining the corresponding recognition accuracy.

PNN is based on the Bayes' theorem, thus the high recognition accuracy close to the theoretical one may be obtained as the number of sample patterns (i.e., neurons in the kernel layer) increases. This is the reason why high recognition accuracy may be obtained in the real world applications. The major difficulty in PNN, however, is that the recognition time (one-shot calculation time in the network) increases as the number of sample patterns increases. Furthermore, the calculation needs to be repeated changing the learning (training) patterns and the smoothing parameter σ. Consequently, it is difficult to finish the learning in real time even using a high performance computer of 1GHz clock speed.

III. ARCHITECTURE FOR HIGH SPEED LERNING

A. SIGMA-PARALLEL ARCHITECTURE(SPA)

We choose the Parzen window function (3) as the kernel function because it is well-suited to the logic circuits. The Parzen window simply means that the hypercube region with sides of length σ is centered on the sample pattern point $\mathbf{S}_j^i$ in the pattern space. The function outputs 1 if the input pattern $\mathbf{X}$ falls within the region. This comparison-based process is easily implemented using simple combinatorial circuits (i.e., AND gates or comparators). Thus, highly parallel neurons can be implemented onto a single or a very few VLSI chips.

For the parallel architecture using multiple processing units, the neuron parallelism seems to be easily applied to PNN, like other hardware implementations of neural networks. In the neuron parallelism (see Fig. 2 (a)), the neurons in the 3 layers are implemented with the specialized circuits in the VLSI chip(s) with memory LSIs: large number of sample patterns for the neurons in the kernel layers are stored in the memory LSIs and those data and kernel function circuits in the VLSI chip(s) construct the neurons in the kernel layer. The difficulty in the neuron parallelism is that the number of neurons in the kernel layer is limited not because of the chip capacity but because of the shortage of I/O pins in the chip. In Fig. 2 (a), q pins for the data width are required for each exterior memory chip. Thus, 'number of memory LSIs' $\times q + \log_2 N_\sigma$ pins are required for the VLSI chip(s).

In order to overcome the above difficulty, we propose the sigma parallel architecture in which every processor calculates the same neuron with the different value of the smoothing parameter σ as shown in Fig. 2 (b). This

Authorized licensed use limited to: Universita degli Studi di Salerno. Downloaded on May 25,2023 at 07:56:10 UTC from IEEE Xplore. Restrictions apply.

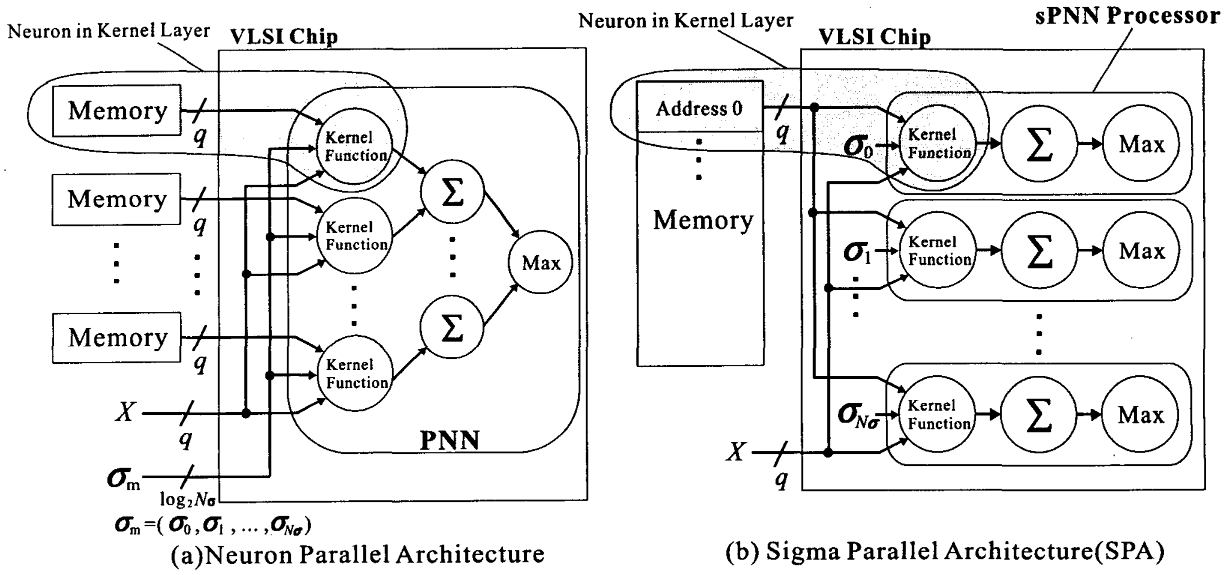

Fig. 2. Neuron Parallel Architecture v.s. Sigma Parallel Architecture.

approach is quite different from the conventional one in which the multiple neurons are calculated in parallel with the same learning parameter (Fig. 2(a)). In the proposed sigma parallel architecture, the sample pattern as well as the input pattern (unknown or test pattern) are broadcast inside the VLSI chip, and every processing unit (we name the unit serial PNN processor, or sPNN processor) calculates the same PNN with the different value of σ. Consequently, only $2q$ pins are required for the multiple processors that work in parallel in the VLSI chip. If the learning is executed with N_σ different values, say N_σ =256, 256 sPNN processors in the VLSI chip work in parallel with 64 (128) I/O pins if q = 32(64) for the floating point precision.

In fact, the neuron parallelism can be combined with the sigma parallelism: Multiple sPNN Processors are got together in a SPA-Module and $N_{Mod.}$ modules are implemented into one VLSI chip, depending on the VLSI chip's total circuit capacity and the number of total I/O pins as shown in Fig. 3. We call this implementation technique Hybrid-SPA.

B. SPEEDUP RATIO

The learning time T_{learn} for one learning is calculated as follows.

$$T_{learn} = \frac{\left(T_{recog.} + \beta N_{sPNN} N_{Mod.} T_{CLK}\right) N_{tp} N_\sigma}{N_{sPNN}}, \quad (4)$$

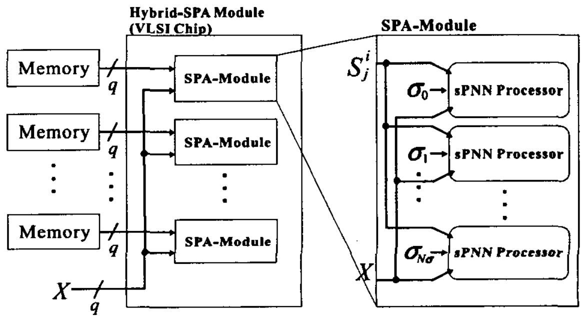

Fig. 3. Neuron Parallelism Based on the SPA.

where

$$T_{recog.} = \left(1 + \frac{nN_pN_c}{N_{Mod.}} + \alpha N_c + \beta N_{Mod.}\right) T_{CLK}, \quad (5)$$

$$\beta = \frac{\log_2 N_p + \log_2 N_c}{s} + \gamma, \quad (6)$$

Authorized licensed use limited to: Universita degli Studi di Salerno. Downloaded on May 25,2023 at 07:56:10 UTC from IEEE Xplore. Restrictions apply.

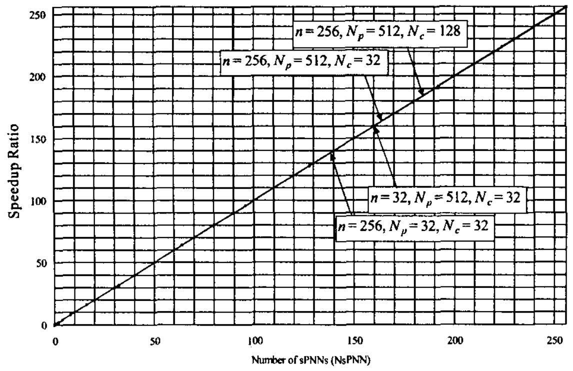

Fig. 4. Speedup Ratio(Scalability Evaluation).

N_{sPNN} :	Number of sPNN processors,
$N_{Mod.}$:	Number of SPA-Modules,
N_c :	Number of categories,
N_p :	Number of sample patterns in each category,
N_{tp} :	Number of test patterns,
N_σ :	Number of σ,
n :	Dimension of the pattern,
s :	Bit width of output data bus,
α :	Overhead cycles for the register setting,
γ :	Overhead cycles for the register setting.

$T_{recog.}$ is time for one recognition, and β indicates overhead cycles for data transfer.

The speedup ratio, or scalability that is calculated as $T_{learn}(N_{sPNN} = 1)/T_{learn}(N_{sPNN})$ is shown in Fig. 4, where the pattern size and its dimension are large enough to a real world image recognition application (i.e., 32 to 128 categories, 32 to 512 patterns/category and test patterns with 32 to 256 dimensions). As shown in Fig. 4, in all cases of the dimension, the number of sample pattern, and the number of category, very high scalability close to the ideal one (i.e. speedup ratio of more than 255.2 at 256 sPNN processors) is obtained even with 256 processors. This result shows that our proposed architecture is very effective in speeding up the PNN learning, while avoiding the implementation problem of the circuits to the VLSI chip.

IV. HARDWARE SYSTEM CONFIGURATION

The hardware configuration of the PNN system that is under development, is shown in Fig. 5. In the development, we use the FPGA chip of Xilinx XCV1000E-6HQ240C that contains the reconfigurable circuits equivalent to 1,569,178 gates[7]. 64 processors (4 data parallelism of 16 sigma parallel each) are integrated into this FPGA chip. Four exterior memory chips (4M-bit SRAM) for pattern data storage are connected to the FPGA. Bandwidth between the FPGA and each memory chip is designed as 8 bits because the system is developed for the image recognition in which the resolution of the image pixel is usually 8 bits.

Each sPNN processor calculates Eq. (3) and outputs the number of kernels that include the input pattern. The output of the processor is set to the register next to it in every category, separately. A Max detector picks up the maximum count of each register as well as its category. The final decision, which is to choose the optimal σ in the learning or the answer (a predicted category) to the input pattern in the recognition, is carried out in another FPGA (Xilinx XCV300E-6PQ240C). These two FPGA chips are connected through the data bus on the board and the data are transferred in the time sharing manner. Thanks to this separative design of chips with the time sharing connection, the number of node processors can be easily increased with simple connection of additional FPGA chips to the bus. The time sharing transfer causes only negligible reduction in the total performance (evaluation results are shown later).

Figure 6 shows a circuit block diagram of the sPNN processor. An adder and a subtracter are for $(s_j^i)_1 + \sigma_m/2$ and $(s_j^i)_1 - \sigma_m/2$ in Eq. (3), respectively and they compose the kernel function. $\left|x_k - (s_j^i)_k\right| \leq \sigma/2$ in Eq. (3) is examined with two comparators sequentially in every component k. The counter increases by one if $\left|x_k - (s_j^i)_k\right| \leq \sigma_m/2$ is satisfied for all components, while the calculation for Eq. (3) is canceled at the RS-FF once it is not satisfied. Finally, the number of kernels that include the input pattern is set to the counter.

V. PERFORMANCE EVALUATION

One sPNN processor is constructed using only 12,000 gates. A controller(not shown in Fig. 5) that controls all processors synchronously in the learning phase as well as in the recognition is also constructed with a small circuit size of 17,000 gates because of the simplicity of the learning algorithm in PNN. The controller thus occupies only 2% of the total circuit area. This simplicity allows 64 sPNN processors to be integrated onto one FPGA chip.

We have developed the PNN hardware prototype (Fig. 7) in order to evaluate the fundamental functions of the sPNN processor. The sPNN Processor Prototype Chip (Xilinx XCS30XL-4PQ240C) shown in Fig. 7 integrates 3 sPNN processors. Two other FPGA chips in the figure are for the best σ detector & PNN controller and for video processing, respectively. The performance of the

Authorized licensed use limited to: Universita degli Studi di Salerno. Downloaded on May 25,2023 at 07:56:10 UTC from IEEE Xplore. Restrictions apply.

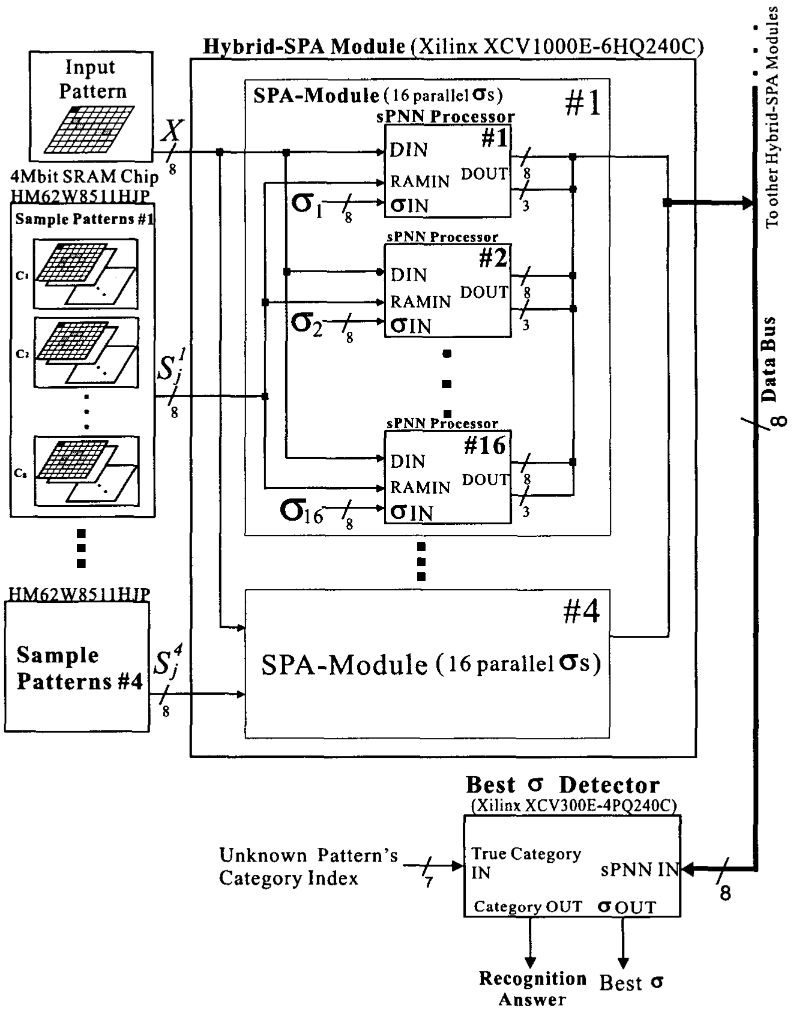

Fig. 5. System Configuration Using the Hybrid-SPA.

sPNN processor prototype chip is evaluated using the preprocessed image data from the video processing chip. All chips operate under the common system clock of 13.5 MHz. After the evaluation, the sPNN Processor Prototype Chip is replaced by the target device (XCV1000E-6HQ240C), and the board is used as the final hardware system.

Measured results showed that one image from the video camera was successfully recognized in 18.96μs. From this fundamental image recognition speed, we have estimated the learning execution time of the PNN hardware system. The result showed that the learning execution time was about 10 seconds under 32 categories and 256 patterns/category, for example. We also measured the execution time for the same learning using a standard PC with a 1-GHz Pentium III processor for comparison. And the PC took longer learning time than that of the PNN hardware by 68 times. The execution time in PC was measured under the standard Linux environment. Thus, the learning time using PC may be able to be shorten further with environmental tuning. However, the PNN hardware is expected to obtain a learning-speed performance that is higher than the PC by 20 times with the environmental tuning at least. This speedup in the learning as well as recognition makes a significant contribution to the large real world applications using PNN.

Authorized licensed use limited to: Universita degli Studi di Salerno. Downloaded on May 25,2023 at 07:56:10 UTC from IEEE Xplore. Restrictions apply.

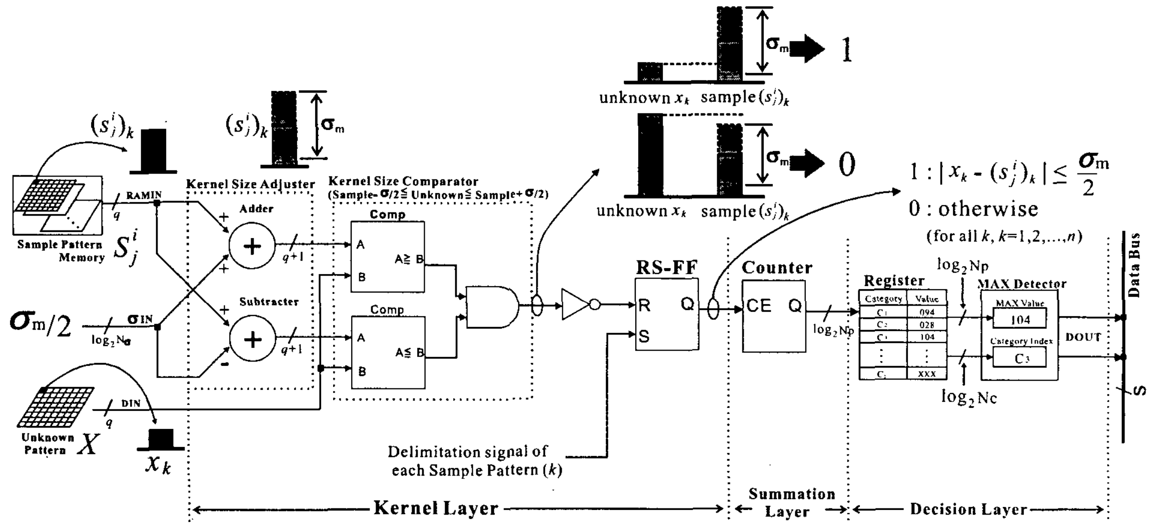

Fig. 6. Circuit Configuration of sPNN Processor.

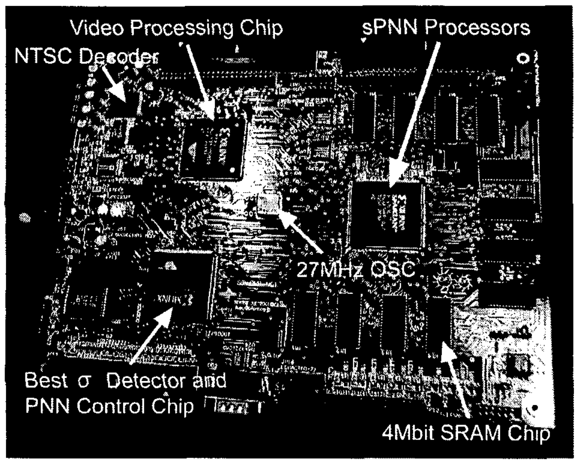

Fig. 7. Photograph of Prototype System.

VI. CONCLUSIONS

The proposed parallel architecture was shown to suit to the PNN hardware, maintaining a high scalability and overcoming the difficulty of the VLSI implementation. The hardware prototype was developed and it executed the fundamental functions of the PNN successfully. From the prototype design and its performance evaluation, it was shown that 64 sPNN processors were implemented onto one FPGA chips (Xilinx XCV1000E-6HQ240C) and the learning execution speed was expected to be faster than that of a standard PC with a Pentium III processor by 20 times at least for a practical application. The proposed hardware system is expected to make a significant contribution to the large real world applications using PNN.

Acknowledgements

This research is supported in part by the 2001 grant of the Ministry of Education, Culture, Sports, Science and Technology in Japan (Grant No. 13450163)

References

[1] D.F.Specht, "Probabilistic Neural Networks," *J. Neural Networks*, Vol. 3, pp.109-118, 1990.

[2] D.F.Specht, "Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification," *IEEE Trans. Neural Networks*, Vol.1, No.1, pp.111-121, 1990.

[3] Bin Tian, Mahmood R.Azimi-Sadjadi, Thomas H.Vonder Haar, and Donald Reinke, "Temporal Updating Scheme for Probabilistic Neural Network with Application to Satellite Cloud Classification," *IEEE Trans. Neural Networks*, Vol.11, No.4, pp.903-920, 2000.

[4] Bin Tian, and Mahmood R.Azimi-Sadjadi, "Comparison of Two Different PNN Training Approaches for Satellite Cloud Data Classification," *IEEE Trans. Neural Networks*, Vol.12, No.1, pp.164-168, 2001.

[5] K.Z.Mao, K.-C.Tan, and W.Ser, "Probabilistic Neural-Network Structure Determination for Pattern Classification," *IEEE Trans. Neural Networks*, Vol.11, No.4, pp. 1009-1016, 2000.

[6] G.Minchin, and A.Zaknich, "A Design for FPGA Implementation of the Probabilistic Neural Networks," *Proc. IEEE ICONIP'99*, pp.556-559, 1999.

[7] Xilinx, Inc."Virtex-E 1.8V Field Programmable Gate Arrays: DS022-1(v2.2)", http://www.xilinx.com/partinfo/ds022.pdf, 2001.

Authorized licensed use limited to: Universita degli Studi di Salerno. Downloaded on May 25,2023 at 07:56:10 UTC from IEEE Xplore. Restrictions apply.