An optoelectronic multilayer network

Alan Yamamura, Seiji Kobayashi*, Mark Neifeld, and Demetri Psaltis

Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125

*Sony Corporation
Tokyo, 100-31, Japan

ABSTRACT

We describe an optoelectronic neural network implementation with electronic neurons and synapses and optical storage of weights and present experimental results.

1. INTRODUCTION

Inspired by biological systems, neural networks may provide the key to solving problems that have long proven difficult using conventional computers. Though neural networks have been demonstrated using a variety of optical and electronic technologies, hybrid optoelectronic implementations combine the strength of optics in communications with that of electronics in computation. Here, we describe a multilayer feedforward neural network implementation using *neurons* (computational elements) and *synapses* (connections between neurons) fabricated on an integrated circuit. Each synapse contains a photodetector thus allowing *weights* governing the connections to be accessed holographically from an optical disk as shown in Fig. 1.

2. THE CHIP

Because of its structure, a multilayer feedforward network can be implemented using a chip with only a single layer of neurons and synapses by repeatedly reconfiguring the synapses before feeding the neuron outputs back to the synaptic inputs. Using this technique, a set of weights must be stored for each layer of every function we would like the network to perform. Since the storage of weights on-chip consumes area that could otherwise be used for computation, we store the weights on an optical disk. Optical storage and readout of the weights provides a number of advantages. The first advantage of optics is the time required to load the weights into the synapses. As the number of neurons N in each layer grows, the number of synpases grows as $O(N^2)$ while the number of pads would at best grow as $O(N)$. Thus $O(N)$ cycles would be required to electronically load the weights for each layer. However with photodetectors in each synapse, the weights for an entire layer can be optically read in parallel via the third dimension. Another advantage of optical media such as the optical disk is that they provide the high density and high capacity storage for the weights required to perform a large number of complex functions.

We have designed and tested an integrated circuit (Fig. 2) containing two layers of 11 neurons and a 15 $\times$ 15 synaptic array connecting them. (For the purpose of electrical characterization, some synapses do not connect to neurons). Each synapse (Fig. 3) contains a synaptic transistor connecting a pair of neurons, one from each layer. The strength or weight of the connection depends on the gate voltage of this transistor. The gate voltage is determined by a pullup transistor to V_{dd} and a reverse-biased photodiode to *ground*. The weight of a synapse is controlled by adjusting the amount of light striking the photodiode. We operate the synapses in a fashion that implements binary (0,1) weights. First, we turn on the pullup transistors with a reset signal that precharges the gates of the synaptic transistors thus setting all weights to 1. Next, we selectively illuminate the photodiodes in some of the synapses, discharging the gates of their synaptic transistors and setting the weights of the selected synapses to 0.

Figure 4 shows the neuron circuit. We can afford to consume more area with each neuron since there only N neurons compared to $O(N^2)$ synapses. The voltage input of each neuron is determined by the output voltages of neurons in the previous layer and the strengths of the synaptic connections to those neurons. To generate its output, each neuron applies a hard threshold to the input voltage by comparing it to an adjustable threshold voltage V_{th}; a cross-coupled inverter is used for the voltage comparison. The output of the voltage comparator is V_{dd} if the input

voltage is above V_{th} and *Gnd* if below, thus providing binary neuron outputs.

The switching energy E_s of each synapse is an important parameter which can be used to determine the optical power required to operate the network at a given speed. The switching energy can be determined using the following equation:

$$E_s = I\ A_p\ t_s \tag{1}$$

where I is the intensity of the light uniformly illuminating the chip, $A_p = 474\mu m^2$ is the area of a photodiode, and t_s is the time required to switch the synapses. Figure 5 shows the current through some synapses as a function of the illumination time for a light intensity of 33.7 $\mu W/cm^2$. By averaging measurements for four intensities, we find a switching energy of .7 pJ. Assuming a switching energy of 1 pJ, we would require 10 mW of incident optical power to run a chip with a 100 × 100 synaptic array at 1 MHz. Operation at this speed requires the use of a memory storage element like the optical disk, capable of data access rates on the order of 10 Gbits/sec.

3. THE DISK

Optical disks are high density, high capacity storage media which, though conventionally read out serially, can provide access to large quantities of data in parallel[1]. Using a prototype Sony system, we have written holograms onto $5\frac{1}{4}$" reflective disks. Data is recorded with 1 μm resolution as variations in the surface reflectivity of the disk yielding a storage capacity of well over 10^9 bits (*i.e.* thousands of images or holograms of a million pixels each) per disk side. Figure 6 shows the reconstruction of a Fourier transform hologram from an optical disk. Depending on the size of the hologram, rotation of the disk over the width of the hologram may be approximated as a simple translation[2]. Because translation of a Fourier transform hologram results in phase variations in, not translation of, the reconstruction, storing the weights as Fourier transform holograms not only allows their parallel optical readout, but also simplifies the alignment of the disk with the chip and provides a significant dwell time of the reconstruction on the photodiodes as the disk rotates. Another advantage of holography is that it represents the desired object in a distributed fashion; thus the reconstruction degrades slowly as the number of defects in the hologram increases. Unfortunately, a large amount of space-bandwidth product (SBP) on the disk must be expended to accurately record and readout the holograms. We expend a factor of 4 in SBP just to place the reconstruction on a carrier to separate it from the bright DC spot. As the number of reconstructed spots increases, we require greater accuracy in the recorded phase, consuming even more SBP. In our holograms, when we devote a factor of 4 in SBP to record the phase with 4 × 1 superpixels, we are only able to reconstruct about 100 points while using a SBP of over 10,000 on the disk.

4. A HETEROASSOCIATIVE MEMORY

We have implemented a two-layer heteroassociative memory using the chip. An input vector (++ ++----+- +-+ -+) is displayed in the bank of LEDs at the top of Fig. 7a. Valid input vectors consist of 8 +1s (on LEDs) and 7 -1s (off LEDs). Each neuron in the second layer acts as a "grandmother-cell" recognizing a specific input vector and associating it with a specified output vector. The first light pattern striking the photodiodes connects each grandmother-cell neuron with only those inputs corresponding to -1s in the specified input vector. With the threshold set to find all -1s, the output of each grandmother-cell will be -1 if and only if the input corresponds to the specified vector. The LEDs on the left of Figs. 7ab show that the second vector (2^{nd} LED off) has been recognized. In order to readout the associated vector, the second light pattern connects each grandmother-cell with output neurons corresponding to -1s in the specified output vector. By setting the threshold to find a single -1, a grandmother-cell with a -1 output forces the outputs of the appropriate output neurons (--++-+-++--) to -1 as shown at the top of Fig. 7b. Since there are 11 neurons in each layer, we were able to store 11 different heteroassocations using the chip. During testing, the network successfully recalled all 11 associations without error.

5. AN AUTOASSOCIATIVE MEMORY

We have also implemented a two-layer autoassociative memory using the chip with 10 input neurons, 5 neurons in the hidden layer, and 10 output neurons. This time, however, instead of using a grandmother cell representation, we use a distributed representation in the hidden layer. We train the two-layer network using a combination of two algorithms. The first algorithm[3] uses learning by choice of internal representation and is designed for use with

multilayer networks of binary threshold elements; the algorithm applies the perceptron learning rule to find the desired analog weights in each layer. The second algorithm[4] is a modified version of the perceptron learning rule that finds binary weights. In our hybrid algorithm, we begin with randomly selected binary weights in the first layer. Presenting the training samples at the inputs, we tabulate the resulting internal representation. We then train the second layer with the binary perceptron algorithm by first applying perceptron learning and then thresholding the resulting analog weights to generate binary ones. If the thresholded binary weights do not operate correctly, we try all neighboring binary weights vectors within a Hamming distance of 1 or 2. If we cannot successfully find weights for the second layer, we modify the internal representation by flipping bits and again apply the binary perceptron to the second layer using the new representation. Once we discover an internal representation for which a corresponding binary second-layer weight vector exists, we search for binary first-layer weights that generate the desired internal representation by applying the same binary perceptron algorithm to the first-layer. If we fail to find the desired first-layer weights, we record the existing internal representation and again start training the second-layer weights, repeating the above process as required.

Despite the fact that the chip provides unipolar weights, we assumed the presence of bipolar weights while training the autoassociative memory since networks with bipolar binary weights have greater functionality than those with unipolar binary weights. By requiring that half the 10 input bits be +1 and half be -1, however, we ensure that operation of the first layer using unipolar weights is functionally equivalent to using bipolar weights. We implement bipolar binary weights in the second layer using a dual-rail coding system by taking the five hidden layer neuron outputs generated by the first-layer and setting five additional hidden layer neuron outputs to the complements of the first. Each output neuron can then connect to either the output of a hidden layer neuron or its complement.

Figures 8ab show the recognition and recall of one of the stored vectors. The input vector (++--++---+) is shown at the top of Fig. 8a. The left side of Fig. 8a shows the corresponding internal representation (++++-) in the hidden layer. The left side of Fig. 8b shows the addition of an externally generated complement to the internal representation (++++- ----+). The top of Fig. 8b shows the successful recall of the stored vector. We stored and recalled six vectors in this 10-5-10 network, again without error.

6. ACKNOWLEDGEMENTS

This research is supported by a grant from the Army Research Office and in part by a grant from the Defense Advanced Research Projects Agency. Thanks to Robert Snapp for his help in working on the binary multilayer network training algorithm. Alan Yamamura is supported by a fellowship from the Fannie and John Hertz Foundation.

7. REFERENCES

1. D. Psaltis, A. Yamamura, M. Neifeld, and S. Kobayashi, "Parallel Readout of Optical Disks," *Third Topical Meeting on Optical Computing*, Salt Lake City, UT, February 1989.

2. D. Psaltis, M.A. Neifeld, A. Yamamura, and S. Kobayashi, "Optical Memory Disks in Optical Information Processing," to appear in *Applied Optics*, 1990.

3. T. Grossman, R. Meir, and E. Domany, "Learning by Choice of Internal Representation," 1989.

4. F. Mok, *Binary Correlators for Optical Computing and Pattern Recognition*, Ph.D. thesis, California Institute of Technology, 1989.

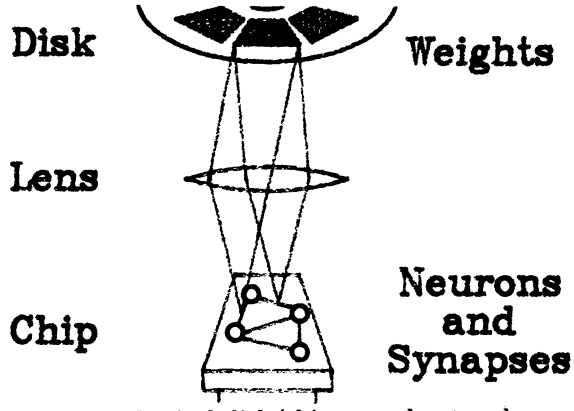


Fig. 1. Optical disk/chip neural network

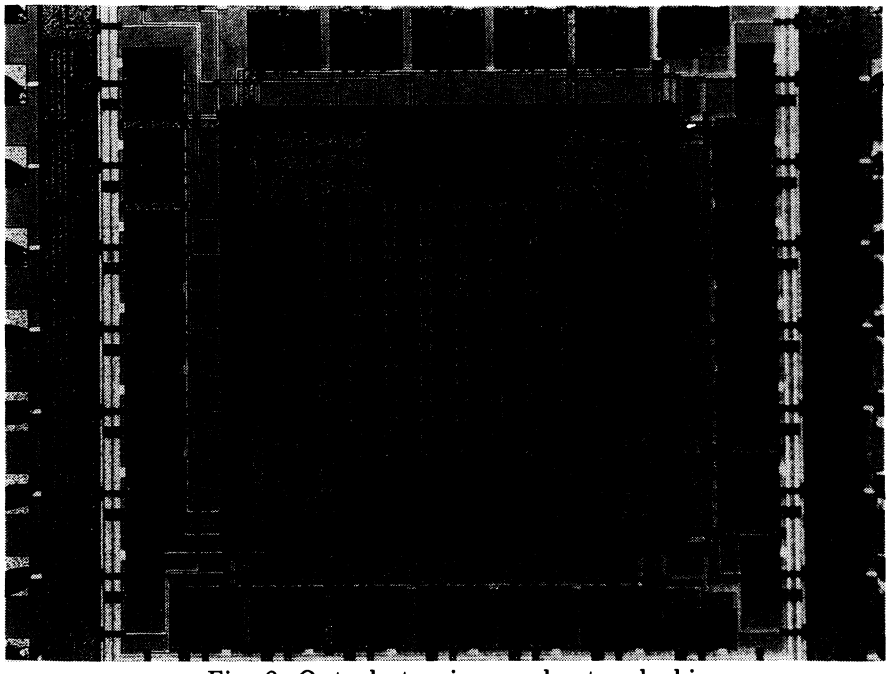


Fig. 2. Optoelectronic neural network chip

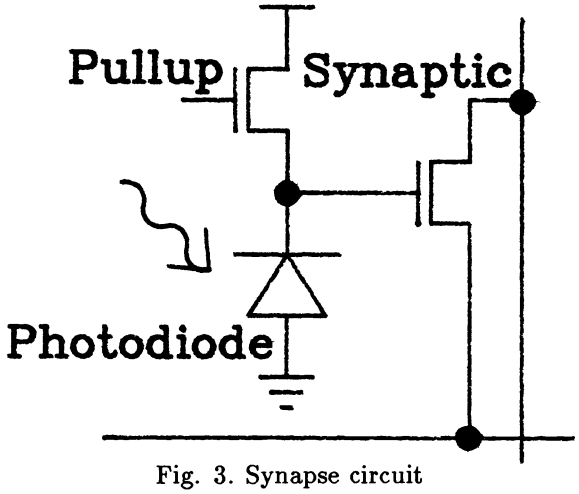


Fig. 3. Synapse circuit

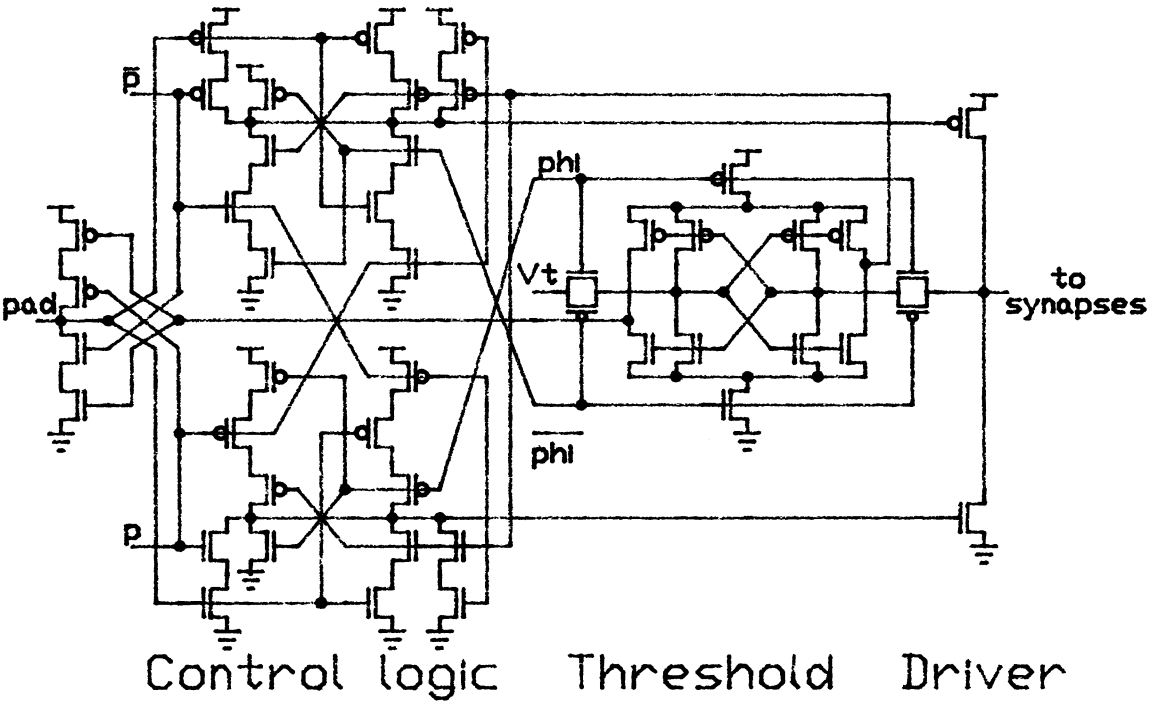
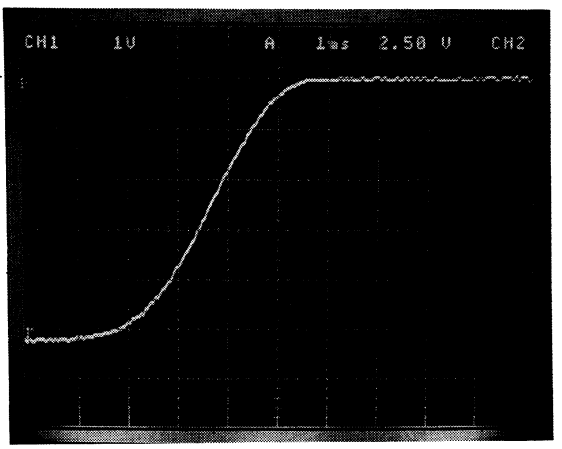


Fig. 4. Neuron circuit



$0\mu A-$

$6.67\mu A/vert.\ div$

$1ms/horiz.\ div$

Fig. 5. Synapse current vs. time ($I = 33.7 = \mu W/cm^2$)

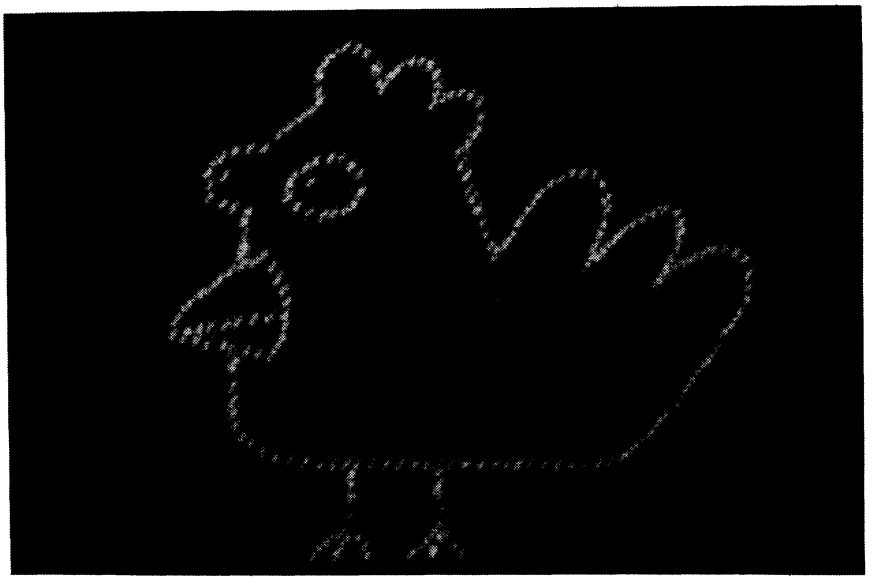
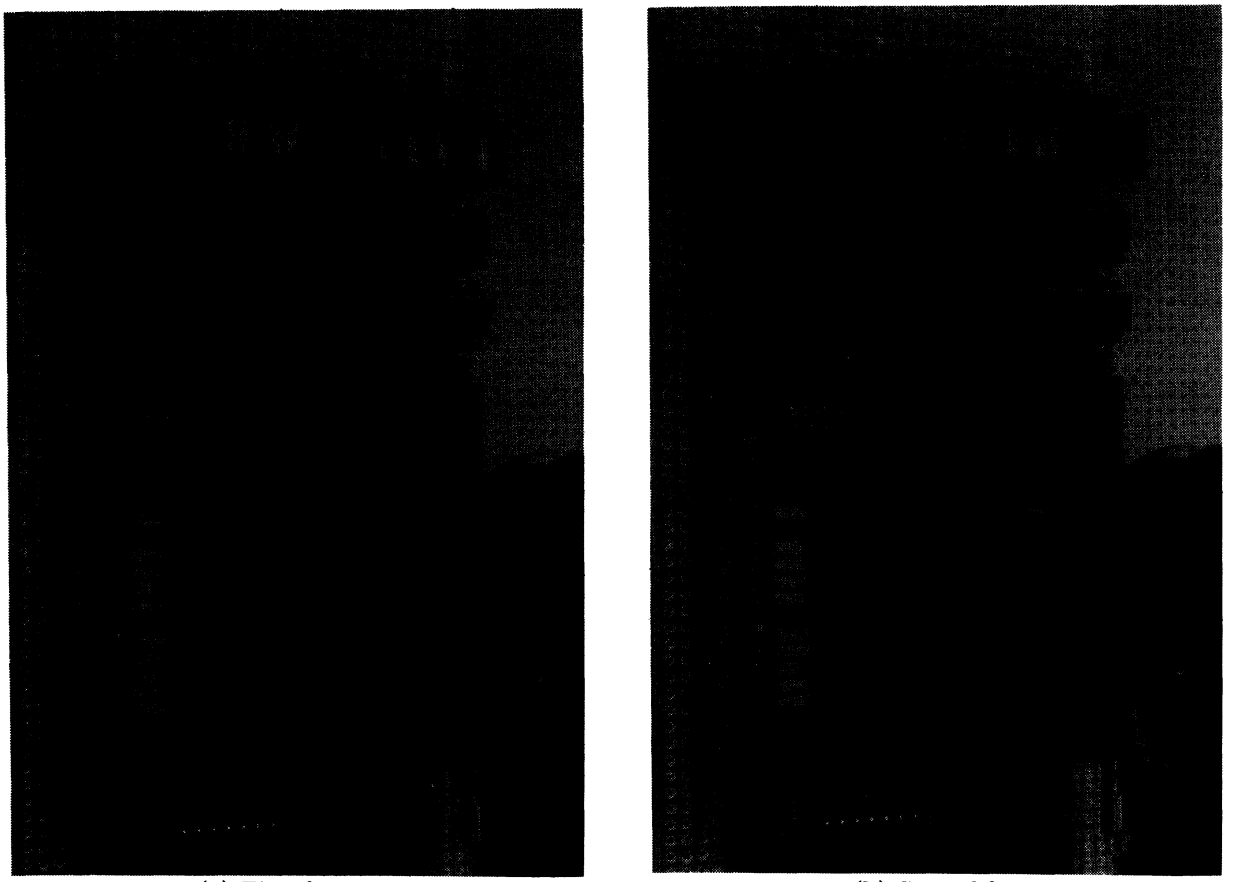
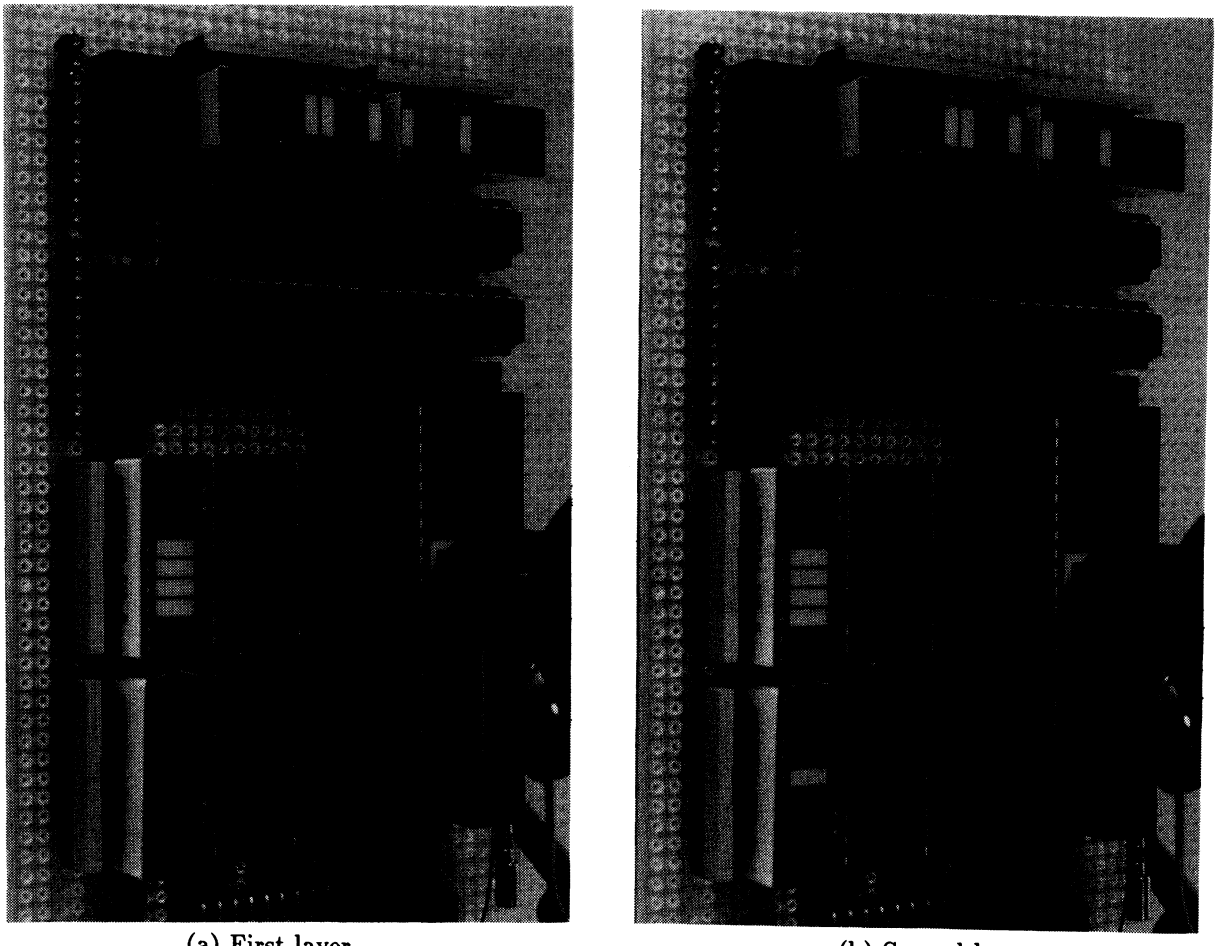


Fig. 6. Fourier transform CGH reconstruction



(a) First layer (b) Second layer

Fig. 7. Heteroassociative memory



(a) First layer (b) Second layer

Fig. 8. Autoassociative memory