Fully-Digital, Highly-Modular Architecture of a Hamming-like Neural Network.

A.Pavlidis[a,b*], A.Arapoyianni[a] and D.Loukas[b]

[a] *University of Athens, Dept. of Informatics*
[b] *NCSR "Demokritos", Inst. of Nuclear Physics*

1 Introduction.

A practical fully-digital network for pattern recognition is presented here. It consists basically of a Hamming-like Network architecture permitting modularity and is suitable for both semi-custom VLSI and FPGA implemantations

A *Hamming Neural Network (HNN)* is a distributed architecture for implementing assosiative memory robust to noise. The stored patterns of such a memory are P vectors of dimesion N, with each vector component taking values -1 or 1 :

$$\begin{gathered}\mathbf{S}^{(k)} = [s_1^{(k)} s_2^{(k)} \cdots s_N^{(k)}]^T : \\ s_i^{(k)} \in \{-1, 1\}, i = 1 \ldots N, k = 1 \ldots P.\end{gathered} \tag{1}$$

When applying a noisy vector at the input

$$\begin{gathered}\mathbf{X} = [x_1 x_2 \cdots x_N]^T : \\ x_i \in \{-1, 1\}, i = 1 \ldots N,\end{gathered} \tag{2}$$

the output of the memory will be one of the stored vectors, this one with the minimum Hamming distance from the applied vector, that is :

$$\mathbf{Y} = \arg\min_{\mathbf{S^{(k)}}} M(\mathbf{X} - \mathbf{S^{(k)}}), \tag{3}$$

where the Hamming distanse is defined as

$$M(\mathbf{X} - \mathbf{S^{(k)}}) = \frac{1}{2}N - \frac{1}{2}\sum_{i=1}^{N} x_i \cdot s_i^{(k)}. \tag{4}$$

The architecture that implements this type of assosiative memory [1] is depicted in Fig.1. It consists of two layers of P neurons. The first layer of the network finds the similarities between the input vector and each one of the stored patterns. The weights and the thresholds for each neuron are defined as $w_{ij} = \frac{1}{2}S_i^{(j)}$ and $\theta_j = \frac{N}{2}$, respectively. In essence, the outputs of the first layer are the crosscorrelations between input and stored vectors. These outputs, $z_j, j = 1 \ldots P$, are called *matching scores* and take values between 0 and N. The second layer of N neurons is a *Winner-Takes-All (WTA)* network that finds by iteration the maximum matching score. This is achieved by assigning the following settings to the second layer weights

$$v_{jk} = \begin{cases} 1, & k = j \\ -\epsilon, & \epsilon < \frac{1}{P-1}, k \neq j \end{cases} \tag{5}$$

After a finite number of iterations [4] the second layer converges and the neuron which corresponds to the most similar stored vector acquires a high value, while the other neurons level off to zero.

2 VLSI implementations.

Several circuit implementations of neural networks have been reported using analog, digital and mixed-mode or hybrid techniques. However, there is not yet a universally dominant approach. The desired characteristics of a VLSI implementation are regularity and modularity in the structural elements of the circuit, local communication between them and low pin-out count. One of the main problems of a neural network implementation is the high interconnection count between the neurons. For the same reason the expandability of the network at the chip level is restricted.

Analog implementations have as structural blocks operational amplifiers or transistors and the information is represented as voltage, current or charge. The main advantages of analog implementations, being profit in processing time, silicon area and power dissipation, cannot always be achieved.

*A.Pavlidis is also grant holder from National Scolarships Foundation

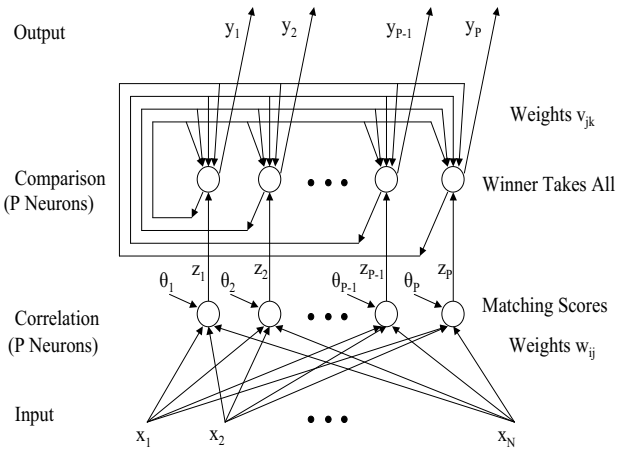

Figure 1: Hamming Neural Network.

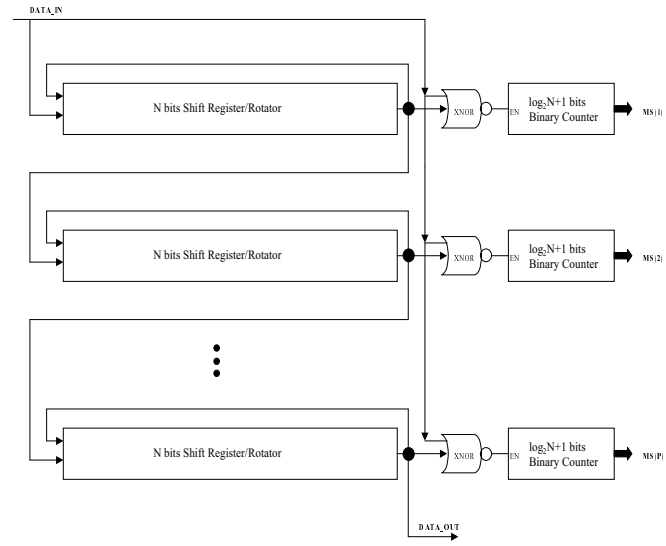

Figure 2: Digital Implementation of the First Layer.

If a direct mapping of the abstract architecture shown in Fig.1 to the circuit level is performed [2] then the high interconnection property is inherited at the circuit level. This leads to an increment of the layout area which is proportional to $O(P^2)$, contradicting the analog implementation advantage of small area. For the same reason the expandability is restricted due to increased processing time and the quadrature complexity in the number of connections. In other attempts for analog implementations [3] characterised by a linear relation of the transistor number relative to the number of patterns, the need for large size of transistors when the number of patterns increases, adds to the size and power dissipartion of the chip. Also, the number of neurons is restricted by the sensitivity to fabrication variations.

In the following section a fully-digital, highly-modular architecture which keeps parameters like area and processing time in good limits is proposed.

3 Digital Implementation of the Assosiative Memory

3.1 First Layer of Neurons.

The correlation of the input vector to the stored vectors can be easily implemented by using XNOR gates as bit-per-bit multipliers (see eq. 5) and binary counters as summing elements. Fig. 2 shows an abstract block diagram of the correlation layer. In each row there are three structures : Shift Register/Rotator of N bits to store the patterns, an XNOR gate and a Binary Counter of $\log_2 N+1$ bits with enable input. The circuit operates in three modes : *Store*, *Correlate* and *Freeze*. The clock is synchronous to the memory elements.

The *Store* mode is used whenever there is need to store new patterns. In this mode the data are clocked serially into the input *DATA-IN* until the first bit reaches the last flip-flop of the last Shifter/Rotator. In this mode the output of every Shift Register feeds the input of the next one. The expandability in the number of the patterns is obvious at this level, as the serial output *DATA-OUT* can be directly connected to the input of the next chip.

In *Correlate mode*, the input pattern is fed to the *DATA-IN* and the Shifter/Rotator acts as a Right Rotator. A parallel bit-by-bit comparison is performed at the XNOR gates for every stored pattern. When the two bits feeding the XNOR are equal the counter is incremented by one (it is assumed that the counter is reset at the previous step). At the end of this step, that lasts N clock cycles, a full rotation has been performed, the stored patterns are at their initial position in the registers, and ready for a new correlation step. The matching scores

appear at the P outputs *MS(1)* ... *MS(P)*.

The *Freeze* mode is used either when the second level of the network is active or when reseting the counter or in idle state, in order to prevent unwanted shift of the stored patterns.

3.2 Second Layer of Neurons.

Having computed the matching scores, the next step is to find the maximum. A modular digital architecture that efficiently finds the winner is described below.

The method succesively checks the bits of the same order (starting from the MSB) of all the matching scores. If some bits are "1" then, the corresponding matching scores are candidate winners, the others with their bit "0" lose and their matching score is set to zero. If all the bits are "0" then nothing is done. In both cases, the procedure applies to the next order bits, until the LSBs. At the end one ore more patterns are the winners.

This procedure needs shift registers for storing the matching scores but instead of that, someone can take advantage of the existing flip-flops in the counters of the previous layer in order to build a Counter/Shift Register shared by the two layers.

An advantage of the above method for finding winners is that it reduces the interconnection complexity to $O(P)$ as it needs only P connections from the Counter/Shift Register to a NOR gate for finding if all the bits are zero while the output of the NOR gate is transmited back to the "neurons" of the second layer. A second advantage of this method is the easy expandability in the number of the patterns by connecting similar chips together. This is because the only information to be passed along the chips is the "all bits zero or not" from the output of the NOR. The time complexity of the second layer as presented above is $\log_2(N) + 2$ clock cycles, and for the case of M chips connected together becomes $M \cdot (\log_2 N + 2)$ clock cycles. The last case is because the output of each NOR gate at every chip is buffered by a flip-flop.

4 Simulation - Results.

This architecture has been designed and simulated in a CMOS 0,8 μm standard cells process. The placement and the routing of the cells has been done manually in order to achieve the optimum layout size. The size of the core chip for 16 pattern of 64 bits is 3.9mm $\times$ 1.4mm, and the layout size of the whole chip is 4.6mm $\times$ 2.1mm. The metalization masks of the layout is shown in Fig.3. Simulations show that a clock frequency of 100 Mhz is achievable.

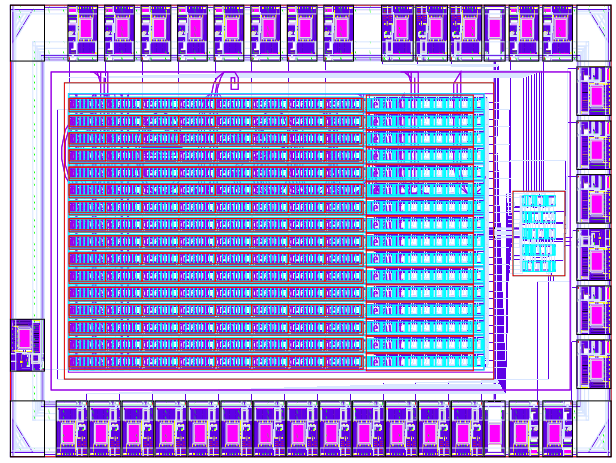

Figure 3: Layout of the designed chip.

Fig.4 shows an excerpt from a simulation of the chip. This excerpt is the comparison described above to extract the winner. In this simulation example the 16 stored patterns with the applied noisy vector appear in Fig. 5. In the simulation timing diagram the signal *MODE* is low indicating the comparison mode of the second level, the signal *SELECT(7:0)* is *0xFF* indicating the Freeze mode of the first layer, and the winner appears at the output *WIN(0:15)* four clock cycles after the signal *MODE* goes low with a value *0x0004* meaning that the pattern 14 won.

5 Conclusions.

A digital architecture implementing a Hamming net has been proposed. A major advantage of this architecture is that beeing fully digital, it can be implemented in FPGA technology attacking thus the high cost of custom analog implementations and the sensitivity limitations inherent in analog designs. The second advantage is that it is highly modular at the chip level, without side effects in terms of speed, as the processing time for each pattern is $N + M \cdot (\log_2(N) + 2)$ clock cycles. Other advantages are robustness and the immunity to the noise, temperature variation and components mis-

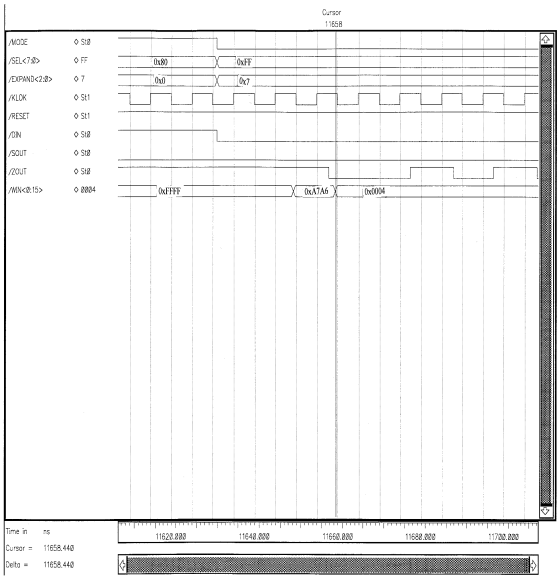

Figure 4: Simulation excerpt of the comparison cycles.

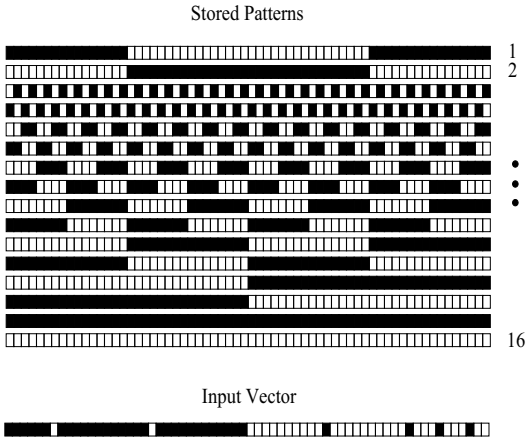

Figure 5: Stored patterns and applied noisy vector for the simulation example.

matches, all these due to the digital structure of the design. Also, it can extract more than one winners, in case of equal Hamming distances, overcoming the problem of convergence in the case of direct implementation of the original architecture in Fig. 1 as reported in [4]. The architecture described in the present summary will be implemented in a FPGA chip for performance assesments. Experimental results will be presented in the conference.

References

[1] R.P.Lippmann. *"An Introduction to Computing with Neural Networks."*, IEEE ASSP Magazine, Vol. 4, pp 8-11, April 1987.

[2] Y.He, U.Cilingiroglou and E.Sanchez-Sinencio. *"A High-Density and Low-Power Charged-Based Hamming Network"*, IEEE Trans. on Very Large Scale Intergration Systems, Vol. I, pp 56-62, March 1993.

[3] M.E.Robinson, H.Yoneda and E.Sanchez-Sinenci. *"A Modular CMOS Design of a Hamming Network.*, IEEE Trans. on Neural Networks, Vol. 3, pp 444-456, May 1992.

[4] K.Koutroumbas, N.Kalouptsidis. *Qualitative Analysis of the parallel and asynchronous modes of the Hamming Network"*, IEEE Trans. on Neural Networks, Vol.5, pp 380-391, May 1994.