Overview of neural hardware

Jan N. H. Heemskerk
Unit of Experimental and Theoretical Psychology
Leiden University, P.O.Box 9555, 2300 RB Leiden
The Netherlands

Draft Version*[1] *Comments are highly appreciated
E-mail: HMSKERK@rulfsw.LeidenUniv.nl
tel: +31 71 273631
fax: +31 71 273619

ABSTRACT

Neural hardware has undergone rapid development during the last few years. This paper presents an overview of neural hardware projects within industries and academia. It describes digital, analog, and hybrid neurochips and accelerator boards as well as large-scale neurocomputers built from general purpose processors and communication elements. Special attention is given to multiprocessor projects that focus on scalability, flexibility, and adaptivity of the design and thus seem suitable for brain-style (cognitive) processing. The sources used for this overview are taken from journal papers, conference proceedings, data sheets, and ftp-sites and present an up-to-date overview of current state-of-the-art neural hardware implementations.

1 Categorization of neural hardware

This paper presents an overview of time-multiplexed hardware designs, some of which are already commercially available, others representing design studies being carried out by research groups. A large number of design studies are being carried out in the US, Japan, and Europe. In many cases these studies concern design concepts of neurocomputers that will never be built in full. Neurocomputer building is expensive in terms of development time and resources, and little is known about the real commercial prospects for working implementations. Moreover, there is no clear consensus on how to exploit the currently available VLSI and even ultra large-scale integration (ULSI) technological capabilities for massively parallel neural network hardware implementations. Another reason for not actually building neurocomputers might lie in the fact that the number and variety of (novel) neural network paradigms is still increasing rapidly. For many paradigms the capabilities are hardly known yet. Paradoxically, these capabilities can only be tested in full when dedicated hardware is available. Commercially available products mostly consist of dedicated implementations of well known and successful paradigms like multi-layer perceptrons with backpropagation learning (e.g., Rumelhart & McClelland, 1986), the Hopfield (e.g., Hopfield, 1982), or the Kohonen models (e.g., Kohonen, 1989). These dedicated implementations in general do not offer much flexibility for the simulation of alternative paradigms. More interesting implementations, from a scientific as opposed to an application viewpoint, can mainly be found only in research laboratories.

Dedicated neural hardware forms the sixth computer generation. The first four computer generations are mainly distinguished by the implementation technology used. They were built up respectively from

[1] This is a draft version based on Chapter 3 in: Heemskerk, J.N.H. (1995). *Neurocomputers for Brain-Style Processing. Design, Implementation and Application*. PhD thesis, Unit of Experimental and Theoretical Psychology Leiden University, The Netherlands. This draft is ftp-able from: ftp.mrc-apu.cam.ac.uk/pub/nn.

vacuum tubes, transistors, integrated circuitry, and VLSI. Third and fourth generation machines were the first parallel machines. Fifth generation computer systems were defined as knowledge based systems, originally designed to accept sensory inputs (Carling, 1992). They formed a combination of AI software and parallel hardware for operations like logical inference and data retrieval. The sixth generation can be seen as an integration of insights into computer design and programming from cognitive science and neuroscience. The machine of the future must perform a cooperative computation system that integrates different subsystems, each quite specialized in structure and some supplied with sensors and motors. A number of these subsystems should implement huge neural networks. Some of the neural hardware implementations described in this overview allow for fast real-world interaction via sensors and effectors and can thus function as action oriented systems (Arbib, 1989).

A large number of parallel neural network implementation studies have been carried out on existing massively parallel machines, simply because neural hardware was not available. Although these machines were not specially designed for neural implementations, in many cases very high performance rates have been obtained. Many examples can be found in the literature, for instance: implementation of backpropagation networks has been performed on the Connection Machine (Singer, 1990), Warp (Pomerleau et al., 1988), MasPar (Chinn et al., 1990; Grasjki, 1992), Hughes (Shams & Gaudiot, 1990), GF11 (Witbrock & Zagha, 1989), AAP-2 (Watanabe et al., 1989), transputer based machines (e.g., Vuurpijl, 1992), and the CRAY YM-P supercomputer (Leung & Setiono, 1993). Much can be learned from these studies about programming neural functions and mapping networks onto pre-specified architectures. Several successful architectural issues have been re-applied in the newer dedicated neurocomputer designs. This paper is limited to an overview of electronic learning neural hardware and will not outline the architectures of general purpose massively parallel computers (supercomputers), or neurocomputer designs based on other (although very promising) implementation techniques such as opto-electronics, electro-chemical, and molecular techniques. Other neural hardware overviews have been written by Treleaven (1989), Nordström et al. (1992), Vellasco (1992), Ienne (1993b), Glesner and Pöchmüller (1994), and Ramacher and Rückert (1991). This neural hardware overview will be given by grouping the different approaches into four main categories, according to a scheme given by Rückert (1993); see Figure 1. Speed performance increases with (bottom) category from left to right.

The first two main categories consist of neurocomputers based on standard ICs. They consist of Accelerator boards which speed up a conventional computer like a PC or workstation, and parallel multiprocessor systems, which mostly run stand alone and can be monitored by a host computer. In these approaches, where standard parts are used, designers can concentrate fully on developing one technology at a time. The other main categories consist of neurocomputers built from dedicated neural ASICs (application specific integrated circuits). These neurochips can be digital, analog, or hybrid. The number of world wide ongoing projects is already too large to allow a complete overview. Special attention will be given to projects which enable the implementation of action oriented systems and supports the following issues: speed, adaptivity, flexibility, and scalability. Reported performance rates are in (million) CPS and CUPS and are taken from publications. They only serve as indications and have to be compared with care since the implementations differ in precision, and size. Furthermore, adequate benchmarks have not yet been generally developed. An often used benchmark for the learning and recall phase of backpropagation networks is NETtalk, which translates text to phonemes (Sejnowksi and Rosenberg, 1987). Other hardware benchmark proposals have for instance been made by Ienne (1993a) and Van Keulen et al. (1994).

2 Neurocomputers consisting of a conventional computer + accelerator board

Accelerator boards are the most frequently used neural commercial hardware, because they are relatively cheap, widely available, simple to connect to the PC or workstation, and typically provided with user

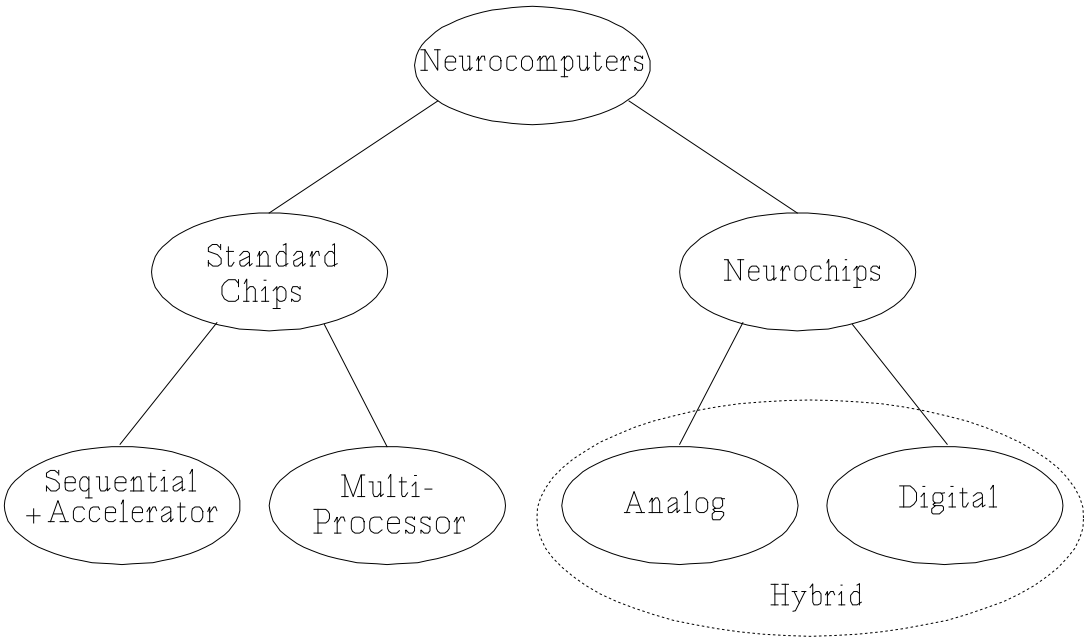

Fig.1. Neurocomputer categories.

friendly software tools. In some cases users are already familiar with the software tools, having used them on their standard, non-accelerated systems. (An overview of neural network simulators can be found in Murre, 1995) Such users will only notice the addition of an accelerator board by the reduced waiting times when training or running a neural model. The speed-up that can be achieved is at about one order of magnitude compared to sequential implementations. Some accelerators come with up to 20 different versions of paradigms which allow for a lot of experimenting. A drawback of this kind of neurocomputer and commercially available software simulators is that they lack flexibility and thus do not offer many possibilities for setting up novel paradigms. Some commercially available accelerator systems will now be briefly reviewed.

ANZA plus (HNC [Hecht-Nielsen Cooperation] CA, USA)
The ANZA coprocessor boards plug in to the backplane of a PC. They contain a Motorola NC68020 processor and a MC68881 floating point coprocessor. Performance rates are given in Treleaven (1989): 1M virtual PEs, 1.5 M Connections, 1.500 CUPS, 6 MCPS. Software: User Interface Subroutine Library.

SAIC SIGMA-1 (Science Applications International Corporation)
This is a Delta floating point processor based system to be plugged into a PC. Performance: 3.1 M virtual PEs and interconnections, 11 MCUPS. Software: neural net library ANSim and ANSpec, an object oriented language (from: Treleaven, 1989).

NT6000 (Neural Technologies Limited, HNC, and California Scientific Software)
Neural Technologies Limited is an organization which offers a wide range of commercially available neural products. The NT6000st and NT6000hs, for instance, are neural network plug-in PC-cards that fulfill the need for intelligent data acquisition. They are equipped with a (TMS320) DSP and an NISP (Neural Instruction Set Processor), which speeds up neural processing to 2 MCPS. Products like these are supplied by well developed software packages that allow for interfacing to other commercially available neural network simulators like BrainMaker and NeuralWorks (see Murre, 1995). The NT6000 series are well suited

to be connected to both analog and digital input/output systems (from: Neural Technologies data sheet, 1993). All this together makes a standard (386 or 486) based PC an interesting neural network implementation tool. The implemented neural network paradigms are, however, limited to backpropagation and Kohonen maps. Little flexibility is left for the development of new paradigms.

Balboa 860 coprocessor board (HNC, CA, USA)
This is a general purpose coprocessor board that can be plugged into an IBM PC but which is also available for Sun workstations (VME-bus). It can be used as an enhancer for the neural software package ExploreNet (also an HNC product). Speed-up to 25 MCPS and 9 MCUPS for the backpropagation paradigm. Its central processor is the Intel i860 chip. This coprocessor board is also used for controlling the HNC SNAP neurocomputer which will be described below (HNC, 1993).

Ni1000 Recognition Accelerator Hardware (Intel Corp., CA, USA)
This recognition accelerator is a PC add-on board that has been specially developed for optical character recognition (OCR) applications. It can be programmed by the NestorACCESS software and mainly uses radial basis functions (RBF) (see Moody & Darken, 1989; Lippmann, 1991) and backpropagation (see Intel, 1992 for an implementation). It gains a speed-up of 100-1000 times compared to implementation on a single DSP or typical host processor.

IBM NEP (IBM Scientific Center, Palo Alto)
This is an experimental neural network programming environment which consist of a Network Emulation Processor (NEP), a cascadable parallel coprocessor for the PC, a Network Interactive Execution Program (IXP), and a high level Generalized Network Language (GNL) (Treleaven, 1989). The hardware consists of a TMS320 digital signal processor and a number of static and dynamic memories, and interfaces to the host PC and a local NEPBUS, which allows for cascading up to 256 processor boards. This forms the so-called Computation Network Environment (CONE) (Hanson et al., 1987). Each NEP can simulate about 4,000 virtual PEs and 16,000 connections, with 30-50 network updates per second.

NBS (Micro Devices, FL, USA)
The MD1220 Neural Bit Slice (NBS) contains 8 digital nodes with 15 'hardwired' parallel synaptic inputs each (Micro Devices, 1990). The binary nodes can process synaptic data in parallel. The number of incoming weights can be expanded to beyond 256 synapses per node, but then additional hardware is required. It can be trained under microprocessor control: no internal learning paradigm is included in the device. Weights are stored externally, and so only off-chip learning is supported. The slice architecture of the NBS permits versatile interconnectivity. Multiple chips can be directly interconnected via parallel I/O links. They are all connected to the CPU bus. The NBS Application Kit contains a PC board with 2 NBS devices and a menu-driven software packet with source. Speed: 55 MIPS (Million Instructions Per Second) per processor, 10 MCPS.

Neuro Turbo I (Nagoya Institute of Technology, Mitec Corp., Japan)
The original Neuro Turbo (Iwata, 1989) consists of 4 DSPs on a board which can be plugged into the extender slots of the NEC personal computer PC98 series. The architecture is used in a successor project (Onuki et al., 1993) which speeds up the original concept by a factor of 2. This uses 16 Fujitsu DSPs (MB68220) arranged in a hypercube.

Neuro Turbo II (MMBB) (Nagoya Institute of Technology, Mitec Corp., Japan)
Arif et al. (1993) describe an accelerator board using general purpose floating point DSPs (Motorola's DSP96002). They used a MMBB (Matrix Memory with Broadcast Bus) and will achieve 16.5 MCUPS. This commercial product can be also seen as the successor of NeuroTurbo I.

Other accelerator board implementations are Parallon and Odyssee (see Treleaven, 1989), and projects described by Panda (1990) and Barber et al. (1993).

3 Neurocomputers built from general purpose processors

General purpose processors, developed for either single processor or multiple processor systems, offer enough programmability for the implementation of neural functions. These implementations will of course never be maximally efficient but because of the wide availability and (in some cases) relatively low price a number of neurocomputers have been assembled from general purpose chips. Implementations range from architectures of simple, low-cost elements (e.g., the BSP400, Heemskerk et al., 1994; COKOS, Speckman et al., 1993) to architectures with rather sophisticated processors like transputers, which are unique for their parallel I/O lines (e.g., Bakkers, 1988; Ernst et al., 1990; Foo et al., 1993; Fujimoto et al., 1992; Grachev and Kirsanov, 1993; Migliore, 1990; Savely, 1987) or DSPs (Digital Signal Processors), which were primarily developed for correlators and discrete Fourier transforms (e.g., Kato et al., 1990; Landsverk, 1992; Onuki et al., 1993). Much experience has been gained from this kind of projects, which can be useful for the design of 'true' neurocomputers (i.e., dedicated neurocomputers completely built from special purpose elements like neurochips). Firstly, such projects confront researchers with a number of problems in the implementation of neuron-like functions. For instance, in many cases the sigmoid function forms the most computationally expensive part of the neural calculation. A solution for this can be found in using a look-up table rather than calculating the function. This is applied in many implementations (e.g., Witbrock & Zagha, 1989; Yasunaga, 1991; Shams & Gaudiot, 1992; Vellasco, 1992). Second, finding an interconnection strategy for large numbers of processors has turned out to be a non-trivial problem. Much knowledge about the architectures of these massively parallel computers can be directly applied in the design of neural architectures. This is important for the design of highly scalable hardware. Many reported designs are kept reconfigurable, offering a testbed for experimenting with several architectures (e.g., RENNS, Landsverk, 1992; Remap, Bengtsson et al., 1993; DREAM machine, Shams & Gaudiot, 1992). Most architectures are however 'regular', for instance grid-based, ring-based, etc. Only a few are hierarchical (e.g., Kotilainen et al., 1993a,b; Ntourntoufis, 1992; Heemskerk et al., 1994). As was argued in Heemskerk (1995), the latter form the most brain-like architectures and therefore receive most attention in the below overview.

WISARD (Imperial College London, UK)
The WISARD systems (see Aleksander et al., 1984) are a series of neurocomputers designed specifically for image processing. They are commercialized as WISARD/CRS1000 by Computer Recognition Systems (CRS) Ltd. (Treleaven, 1989). WISARD (Wilke, Stoneham, Aleksander Recognition Device) systems are based on RAM arrays and therefore limited to a special image recognition process. The input image is divided in tuples of n bits that form the addresses of the RAM (discriminator) cells. All addressed cells are filled with ones, the others contain a zero. The recognition of another image is determined by the number of RAM cells that contain a one.

Mark III and IV (TRW-HNC, CA, USA)
Mark III was the earliest commercially available neurocomputer. The architecture of the Mark III (Hecht-Nielsen, 1990) consists of a broadcast bus. Eight Motorola 68010-based module boards are connected to the bus. The design of Mark III and IV is based on replicative accelerator boards. The Mark III implements 8000 virtual nodes with 80,000 connections. It is connected to a host computer (DEC VAX). The software for these Mark computers is also a product of the HNC and is called Artificial Neural System Environment (ANSE). Performance (from Treleaven, 1989): Mark III: up to 15 physical processors, MC68020 + MC68881. 65,000 virtual processor elements, over 1,000,000 connections, 0.45 MCPS; Mark IV: 236,000 virtual PEs. 5.5 M connections, 5 MCUPS.

Sandy/8 (Fujitsu Laboratories, Japan)
This machine is a prototype for a systolic array architecture proposed by Kato et al. (1990). 256 TMS320C30 DSPs are connected to 'trays' (or communication elements) which are specially designed to implement a circular shift register communication structure. Each tray is connected to its two neighbors and functions as container and router for the data to be transmitted. The prototype is connected to a Sun-3 host machine. Estimated performance on backpropagation: 135 MCUPS.

GCN (Sony Corp., Japan)
In this GigaCoNnection system the 128 PEs are based on the i860 core and they are arranged in a 2-dimensional grid. Expected performance: 1 GCUPS (with training pattern parallelism), Hiraiwa et al. (1990).

Topsi (Texas Instruments, USA)
This is a DSP based architecture with 100-1000 modules consisting of a DSP and a general processor plus communication hardware (Nordström & Svensson, 1992). The modules communicate via a general reconfigurable inter module network, but broadcasting via an additional bus is also possible. Expected performance: 150 MCUPS - 1.4 GCUPS.

BSP400 (Leiden University, Delft University of Technology, The Netherlands)
This experimental neurocomputer was built by us (see also: Heemskerk, 1995, Heemskerk et al., 1991, 1994; Hoekstra et al., 1990; Hudson et al., 1991). We will now cite a BSP400 description by Kotilainen et al. (1993b). "The BSP400 utilizes 400 commercial low-cost microprocessors organized in clusters of 16 processors. The clusters are connected together via a parallel communication bus. The communication bus is used for data transfers between the processor clusters and a common network controller. The philosophy of the BSP400 system is to use larger number of slow low-cost processors operating in parallel rather than designing fast and advanced hardware for individual nodes. Each processor operates on 1 MHz clock frequency and contains 2 Kbytes of EPROM and 128 bytes of RAM. Each cluster of 16 processors is equipped with a control logic block and an input activation queue. The network controller is a small self-contained computer with several parallel I/O lines and a serial port. The controller is connected to a host computer via the serial port. The task of the controller is to distribute the input activation values to the processing clusters and monitor the convergence of the network algorithm. The BSP400 is programmed by loading the code containing proper activation and learning rules to the EPROM's of the individual processors. The 128 bytes of RAM is used for storing the adaptable weights of the networks as well as other parameters and variables needed in calculations. The performance rate is 6.4 MCPS." In a following up study the mapping of large-scale neural topologies on an extended BSP-like architecture are simulated and analyzed. The successor to the BSP400 will be called MindShape and uses a hierarchical bus-oriented Fractal Architecture. MindShape stands for 'Modular Interconnected Neural Design with Scalable Hardware and Adaptive Processing Elements'. Publications on the MindShape concept can be found in: Heemskerk et al. (1992), Heemskerk (1995), and Heemskerk and Murre (1995).

DREAM Machine (Hughes Research Laboratories, CA, USA)
In the DREAM Machine (Dynamically Reconfigurable Extended Array Multiprocessor, see Shams and Gaudiot, 1992) special processors with local memory are arranged on a 2D lattice with eight nearest neighbor connections implemented through dynamically reconfigurable switches. The sigmoid is implemented by a look-up table. Mapping is done by embedding a variable length computational ring on a 2D topology to get a 'snake-like' 1-D ring. One to one implementations are most efficient. Two fully interconnected layers are embedded on the same ring and can exchange their activations in a systolic way. A backpropagation (NetTalk) implementation on a 256 nodes DREAM machine performed 517 MCPS.

RAP (ICSI, Berkeley, CA)
The RAP (Ring Array Processor, Morgan et al., 1992) was developed at the ICSI (International Computer Science Institute) and has been used as an essential component in the development of connectionist

algorithms for speech recognition since 1990. Implementations consist of 4 to 40 Texas Instruments TI-TMS320C30 floating point DSPs containing 256 Kbytes of fast static RAM and 4 Mbytes of dynamic RAM each. These chips are connected via a ring of Xilinx programmable gate arrays (PGAs), each implementing a simple two register data pipeline. Additionally each board has a VME bus interface logic, which allows it to connect to a host computer. The software support of RAP contains a workstation based command interpreter, tools for the standard C environment and a library of matrix and vector routines. A single board can perform 57 MCPS when computing a multi-layer perceptron network in forward operation, and 13.2 MCPS with backpropagation training.

COKOS (University of Tübingen, Germany)
A CO-processor for Kohonen's Self-organizing feature map built from off-the-shelf components and floating point gate arrays (see Speckman et al., 1993). A prototype has been built and consists of 8 Memory and Arithmetic Boards (MABs), an adder tree and a controller. The system is connected to a host PC via an asynchronous link. The PC runs special software for initiating the co-processor and analyzing the data. Performance: 16 MCUPS (Kohonen learning steps). This is about 16 times faster than an implementation on a SUN 4.

REMAP (Luleå University of Technology and Halmstad University, Sweden)
This project is aimed at obtaining a Massively Parallel Computer Platform for Neural Computations (Bengtsson et al., 1993) using field programmable gate array (FPGA) technology. The architecture of REMAP (Real-time Embedded Modular Adaptive Parallel processor project) can be adjusted to a specific application. This reconfigurability is the most significant property of REMAP. It is a SIMD processor array where PEs are RAM-based field programmable gate arrays. Bit-serial arithmetic is used with a hardware multiplier. The planned system will consist of several computing modules, each being a SIMD computer itself. Estimated performance: 100 MCPS - 1000 MCPS. Implementation of multi-layer perceptron (MLP) with backpropagation, Kohonen's self-organizing maps and sparse distributed memory.

General Purpose Parallel Neurocomputer (Tampere University, Finland)
This general purpose neurocomputer will operate as a neural co-processor. The overall structure consists of a number of identical (TMS320C25 DSP based) processing units, a tree shaped network of communication units, and a single host interface unit. The communication network is used to transfer the data between the processing units and the host interface using broadcast, read, and write operations. Additionally, the communication units can be adjusted such that the network performs global computation tasks. These tasks include, for instance, computing the sum of all values sent from the processing units and determining the minimum or maximum values. Parallelization on the processing units can be done on several levels (layer, node, synapse). The processing units can be equipped with A/D and D/A convertors to provide analog interfacing with the external world. The designers of this neurocomputer have shown the suitability of this architecture to the mapping of MLPs and backpropagation learning, the Kohonen self-organizing feature map (SOFM), and Kanerva's sparse distributed memory (SDM) (see Kotilainen et al. 1993a,b).

TI NETSIM (Texas Instruments and Cambridge University, UK)
A NETSIM system consists of a collection of neural network emulator cards physically connected in a 3-dimensional array with a PC host as front end (Garth, 1987). A NETSIM card is based around a 80188 microprocessor, some memory, a solution engine, and a communication processor. The solution engine has four basic instructions and consists of a math processor, an address controller, and a memory controller. Communication is performed via message passing. Up to 15 NETSIM cards can be addressed in each of the three dimensions. Each card implements 256 nodes with 256 synapses per neuron. The NETSIM neurocomputer supports the majority of common neural network paradigms. The overall performance is 450 MCS and 90 MCUPS.

GeNet (IBM Research Division, Munich, Germany)
GeNet (Generic Network) is a SIMD massively parallel digital neural network simulator, with a highly scalable hierarchical architecture (Soegtrop and Klagges, 1993). The neural core operations are supported by optimized and balanced computation and communication hardware that sustains heavily pipelined processing. In addition to an array of processing elements with one global bus, the system is equipped with a ring shifter (32 bit) and a parallel vector bus that feeds a tree-shaped global vector accumulator. The tree-like I/O bus of GeNet installations may be dynamically subdivided into multiple I/O-busses. Performance: 1.7 GCPS (256 physical nodes).

4 Neurocomputers built from neurochips

In the projects described above the neural functions are programmed on general purpose processors. In special purpose chips dedicated circuits are devised for the neural functions. This will speed up the neural iteration time by about 2 orders of magnitude compared to general purpose processor implementations. Several implementation technologies can be chosen for the design of neurochips. The main distinction lies in choice of a fully digital, fully analog, or hybrid design. Direct implementation in circuits in many cases alters the exact functioning of the original (simulated or analyzed) computational elements. This is mainly due to limited precision. The influence of this limited precision is of great importance to the proper functioning of the original paradigm. Many chip designers seem at the moment to be more concerned about these topics than about the other important issue of the neural chip integration into large-scale architectures. In order to build large-scale implementations, many neurochips do have to be interconnected. Some chips are therefore supplied with special communication channels. Other neurochips are to be interconnected by specially designed communication elements. Several interconnection architectures are presented, some based on earlier experience with multiprocessor designs. Section 4.1 describes some fully digital implementations; Section 4.2 presents some fully analog implementations. The combination of these implementation technologies is described in Section 4.3, which deals with hybrid systems.

4.1 Neurocomputers built from digital neurochips

Digital Neural ASICS (Application Specific Integrated Circuits) are the most powerful and mature neurochips. The first commercially available neurochip was marketed in 1989 by Systonic Systems Beaverton and called Dendros I (Schwartz, 1990). Digital techniques offer high computational precision, high reliability, and high programmability. Furthermore, powerful design tools are available for digital full- and semi-custom design. Disadvantages are the relatively large circuit size compared to analog implementations. Synaptic weights can be stored on or off chip. This choice is determined by the trade-off between speed and size.

SYNAPSE-1 (Siemens AG, Corp, R&D, Germany)
SYNAPSE-1 (synthesis of neural algorithms on a parallel systolic engine) is a general purpose neurocomputer built by Siemens AG (see for instance Beichter et al., 1993; Ramacher, 1991; Ramacher et al., 1993). It consists of a 2-dimensional systolic array of processors, arranged in two rows by four columns. The processors are specially developed MA16 neural chips, which directly implement parts of network formula that are shared by all paradigms, e.g., matrix-by-vector multiplication and maximum finding. Henceforth the MA16s are called Neural Signal Processors (NSPs). All other computations which are uncritical in terms of computational throughput are realized by off-the-shelf hardware. The weights are stored off chip in local memories. Both processor rows are connected to the same weight bus, thus executing the same neural net operations for different input patterns. The MA16s in a row form a linear systolic array where input data as well as intermediate results propagate to obtain the total weighted sums. Each MA16 itself contains a linear systolic array of 4 processing modules, each containing 16x16 bit multipliers. The data

paths and the processing modules can perform neural algorithms as well as classical signal processing algorithms. Multiple standard 68040s with additional integer ALUs are used as general purpose processors (Data Unit and Control Unit) to complement the systolic processing array. SYNAPSE-1 is connected to a Host workstation through the Control Unit. The SYNAPSE-1 standard configuration has eight MA16s, two MC68040 CISC processors for control purposes, and a 128 Mbyte DRAM bank. A neural Algorithmic Programming Language, embedded in C++ has been defined. The prototype, with 8 MA16s performs 5.1 GCPS (and 33 MCUPS), at a clock rate of 40 MHz. A commercial version of the SYNAPSE-1 system has been marketed since September 1994.

CNAPS (Adaptive Solutions, Inc., USA)
The CNAPS (Connected Network of Adaptive Processors) from Adaptive Solutions is a SIMD array with 64 processing elements per chip (Hammerstrom, 1990; Hammerstrom & Means, 1990; Hammerstrom & Nguyen, 1991). The standard system consists of four chips on a board controlled by a common microcode sequencer chip. All processors receive the same instruction which they conditionally execute. The 64 processing elements are implemented in the specially designed N6400, which are comparable to DSPs with reduced precision multipliers. The N6400 chips are connected in cascade and communicate with a broadcast interconnection scheme. The chips have on board SRAM sufficient to hold 128K 16 bit weights. This supports on-chip learning. There is no support for off-chip memory, so that chips must be added when larger networks are to be implemented. Each N6400 contains 64 processing units and 256 Kbytes weight memory (1-16 bits resolution). A single chip can perform 1.6 GCPS (256 MCUPS, backpropagation) for 8 or 16 bit weights and 12.8 GCPS for 1 bit weights. The complete CNAPS system consists of a CNAPS server which connects to a host workstation, and Codenet, a set of software development tools. It supports Kohonen LVQ (linear vector quantization), backpropagation, and convolution at high speed. The developers claimed in 1994 that the CNAPS system was the world's fastest commercially available learning machine. Performance of the CNAPS server with 512 nodes (from: Adaptive Solutions, 1993): 5.7 GCPS, 1.46 GCUPS (backpropagation).

SNAP (HNC, CA, USA)
The SIMD Neurocomputer Array Processor is based on the HNC 100 NAP (Neural Array Processor) chips, each consisting of a 1-dimensional systolic array of four arithmetic cells. The organization of the processing elements is similar to that of the CNAPS system where the chips contain 80 cells. A SNAP board contains four HNC 100 NAP chips. The boards can be connected to the Balboa 860 acceleration board that was described above. A standard configuration of 2 SNAP boards (32 processors) performs 500 MCPS and 128 MCUPS (on backpropagation, see HNC, 1993).

CNS Connectionist Supercomputer (ICSI, Berkeley, CA, USA)
The Connectionist networks supercomputer CNS-1 is the successor of the RAP and is in its design phase (Morgan et al., 1992; Morgan, 1993; Asanovic et al., 1993). Once built it will supply orders of magnitude more capability than the RAP. It is designed for connectionist network calculations and other moderate precision integer calculations.

Custom VLSI digital processing nodes (named Torrent) are connected in a 2D mesh topology and operate independently in an MIMD (multiple instruction stream, multiple data stream) style. Each Torrent has a vector coprocessor which accelerates neural computation, a scalar RISC unit for non-neural calculations, and contains a private memory space (DRAM). The nodes communicate with each other through a simple message-passing scheme. The router is also implemented on the chip. The CNS design is scalable to 1024 processing nodes, which gives a computational capacity of 2 TeraOps (integer operations per second) and 32 Gbyte of RAM. One edge of the mesh is reserved for attaching I/O devices, allowing up to 8 Gbytes/s of I/O bandwidth. The target performances are: 10^{11} CPS and 3^{10} CUPS, which is about 10,000 times faster than a RISC station. There are plans to connect the disk drives directly to the mesh in order to support fast mass-storage. The CNS-1 array is connected to a host workstation. The largest systems supported will have 8000 datapaths operating at 125 MHz.

Much effort is being put into developing the software environment of the CNS. It is argued by the designers of the CNS that raw speed is only useful when the user can program in a general purpose language (C and C++) in a general purpose programming environment, rather then working with point and click actions in neurally oriented interfaces. The latter can of course simplify demonstration of toy problems, but typically are not sufficient for the general kind of programming that users require for research in a complete application. As a solution to this problem they have implemented neural operations such as matrix-vector multipliers in assembly code on the neurocomputer. These routines can be simply called by the other (non-neural) code that is also running on the neurocomputer. Performance goals: 100 GCPS.

Hitachi WSI (Hitachi Central Research Laboratory, Kokubinji Tokyo, Japan)
Prototyping of this Wafer Scale Integration architecture was finished in 1990 (see Yasunaga et al., 1991) and it is still (1995) the fastest existing implementation of backpropagation learning in the world (2.3 GCUPS). Nevertheless Hitachi is not putting it into production (but see MY-NEUPOWER in the next section). 144 digital neurons are integrated on a 5-inch silicon wafer (0.8 μm CMOS technology). Eight wafers are connected by a time-sharing hierarchical bus architecture. Wafer scale integration yields a higher density and allows higher speed than ordinary VLSI technology, but has the disadvantage of a higher percentage of manufacturing defects. It is argued that neural networks are a good candidate for WSI because of the inherent graceful degradation properties which will compensate for this kind of error. Some extra redundancy is put in the design for circuits such as bus wires, bus drivers and nonlinear function tables. The hardware can be connected to a host workstation. For the implementation of the backpropagation learning algorithm a dual-network architecture is used whereby two physical nodes represent one backpropagation node. This allows for performance of the forward phase (where activations are broadcast) and the backward phase (where errors are broadcast) independently. The learning circuits, which take a large amount of chip space, are shared by the neurons. Performance: 1152 nodes, 2.3 GCUPS, with 8 WSI boards.

MY-NEUPOWER (Hitachi Microcomputer System, Ltd., Tokyo, Japan)
This commercially available neurocomputer is based on the experiences gained in building the WSI prototype as described above. It uses specially developed LSIs, and can support various learning algorithms: backpropagation, Hopfield, LVQ (from data sheet Hitachi, 1993; Sato et al., 1993) and contains 512 physical neurons. Larger networks can be simulated by assigning the neurons to the global memory of NEUPOWER. The number of virtual neurons is 4096 for backpropagation. Performance 1.26 GCUPS, 12.8 GCS. My-Neupower can be used as the engine for the software packet NEUROLIVE (a neural network simulator also developed by Hitachi).

LNeuro 1.0 (Neuromimetic Chip, Philips, Paris, France)
LNeuro (Learning Neurochip) is a general purpose digital neurochip which implements 32 input and 16 output neurons (see: Duranton & Sirat 1989; Theeten et al., 1990; Mauduit et al., 1992). Only the loop on input neurons is parallelized, which means that the whole set of synaptic weights related to a given output neuron is updated in parallel (weight parallelism). Its fully digital design offers the undeniable benefit of a known precision, well defined states, and full control of the parameters. The chip comprises on-chip learning with an adjustable learning rule. A number of chips can be cascaded within a reconfigurable, transputer controlled network. In Mauduit et al. (1991) experiments with 16 LNeuro 1.0 chips are described. This machine can simulate networks of more than 2000 nodes. An implementation with a single LNeuro yielded a speed-up of eight compared to an implementation on a transputer. The authors guarantee a linear speed-up with the size of the machine. The next (Tnode) machine will consist of 100 LNeuros. Measured performance: 16 LNeuros on 4 dedicated boards: 19 MCPS, 4.2 MCUPS.

UTAK1 (Catalunya University, Spain)
The Unit for Trained Adaptive Knowledge (see Castillo et al., 1992) is a chip with a single neuron. Multiple UTAK1s can be connected to form a dynamic ring architecture. Together with a controller this can be connected to a PC.

GNU Implementation (Imperial College, London, UK)
This is a parallel hardware implementation of the GNU (General Neural Unit; see for instance Aleksander, 1990). The proposed architecture (in: Ntourntoufis, 1992) consists of 8 rings which are themselves connected to a major ring. The architecture is said to involve a compromise between a ring and a 2-dimensional mesh. This hardware is specially developed for GRAM-based (Generalizing Random Access Memories) systems as are used in the WISARD. Each ring consists of 8 processing elements implementing the functionality and the memory of (typically 16) GRAMs. The rings form a message passing MIMD. This approach forms a typical example of a hierarchical architecture.

UCL (Esprit Pygmalion/Galatea Project at University College London, UK)
A VLSI Architecture for Neural Network Chips (Vellasco, 1992) is proposed which is part of the Esprit II Pygmalion/Galatea project. The aim of this project is to develop a neural network VLSI design environment where a neural application can be automatically mapped from a high-level programming environment into custom VLSI. A prototype backpropagation chip has been designed according to the so-called generic neuron model (see Vellasco, 1992). This allows for node parallel implementation, which does fulfill the demands for flexibility and scalability according to its designers. The chips consist of a RISC processor, a communication unit and memory. For communication a ring structure and a broadcast bus are used.

Mantra I (Swiss Federal Institute of Technology)
The designers of the Mantra I aimed at a multi-model neural computer which supports several types of networks and paradigms. It consists of a 2-D array of up to 40x40 GENES IV systolic processors and the linear array of auxiliary processors called GACD1 (Ienne, 1992; Ienne & Viredaz, 1993; Viredaz, 1993). The GENES chips (Generic Element for Neuro-Emulator Systolic arrays) are bit-serial processing elements that perform vector/matrix multiplications. The Mantra architecture is in principle very well scalable. Performance: 400 MCPS, 133 MCUPS (backpropagation).

Biologically-Inspired Emulator (Duisberg, Germany)
This approach differs from many others in that it allows for the implementation of very complex neurons (Richert et al., 1993). These 'biological' neurons incorporate not only synaptic weighting, postsynaptic summation, static threshold, and saturation, but also synaptic time delays, neuron gain, computation of membrane potential, and dynamical thresholding. All these characteristics are variable and learnable, yielding an opportunity to simulate very sophisticated neurons. The emulator consists of neuron chips (with about 20 'biological' neurons) and communication elements that will be interconnected in a grid array (16 nodes on a board). This highly flexible emulator allows for the implementation of dynamic time-varying architectures which might be of considerable value for future brain research. An example of a time-varying model is described by Eckmiller and Napp-Zinn (1993).

INPG Architecture (Institute Nationale Polytechnique de Grenoble, France)
The neural processors consist of I/O registers, memory, and some calculation logic (Ouali & Saucier, 1990). Each chip has a High and a Low bus. The final architecture is formed by grouping many processors into a 2D array. The busses are made up of bus segments which are linked by software programmable switches. These can be programmed by the PEs during execution according to the neural computing phase being performed. More on this architecture can be found in Ouali et al. (1991).

BACHUS (Darmstadt University of Technology, Univ. of Düsseldorf, Germany)
Three versions of BACHUS exist. A BACHUS III chip contains the functionality of 32 neurons with 1 bit connections. In the most sophisticated version eight BACHUS III chips are mounted together resulting in 256 simple processors. The total system was called PAN IV. Chips are only used in the feed forward phase; learning or programming is not supported and thus has to be done off-chip. The system only supports neural networks with binary weights. Applications are to be found in fast associative databases in a multi-user environment, speech processing, etc. (Glesner and Pöchmüller, 1994).

ZISC036 (IBM Micro electronics, France)
The Zero Instruction Set Computer (ZISC036) neurochip is the first element in a series from IBM of fully integrated implementation of RBF-like (see Moody & Darken, 1989; Lippmann, 1991) neural networks (Eide et al., 1994). The chips have been designed for cost-effective recognition and classification in real-time. It comprises 36 simple neurons with local memory and registers for RBF-type data storage. The chips are cascadable allowing the implementation of larger networks. Communication between ZISCs is achieved through communication busses. ZISCs can be easily interfaced with existing hardware such as PCs.

4.2 Neurocomputers built from analog neurochips

Analog electronics have some interesting characteristics which can directly be used for neural network implementation. Operational amplifiers (Opamps), for instance, are easily built from single transistors and automatically perform neuron-like functions, such as integration and sigmoid transfer. These otherwise computationally intensive calculations are automatically performed by physical processes such as summing of currents or charges. Analog electronics are very compact and offer high speed at low energy dissipation. With current state-of-the-art micro electronics, simple neural (non-learning) associative memory chips with more than 1000 neurons and 1000 inputs each can be integrated on a single chip performing about 100 GCPS.

Analog neural ASICS can operate in either discrete or continuous time. Disadvantages of analog technology are the susceptibility to noise and process-parameter variations which limit computational precision and make it harder to understand what exactly is computed. Chips built according to the same design will never function in exactly the same way. However, this could also be said about the 'wetware' of biological brains and it could be argued that in real brain-style implementations these problems will be overcome by the neural paradigm itself. Real brains keep going in a relatively wide temperature range as well!

Apart from the difficulties involved in designing analog circuits, the problem of representing adaptable weights is limiting the applicability of analog circuits. Weights can for instance be represented by resistors, but these are not adaptable after the production of the chips. Chips with fixed weights can only be used in the recall phase. They are for instance implemented as associative memory chips (e.g., Jackel et al., 1987). Implementation techniques that do allow for adaptable weights are: capacitors (Schwartz, 1990), floating gate transistors (Holler et al., 1989), charge coupled devices (CCDs) (Agranat et al., 1990; Hoekstra, 1990), cells with ferro-electric films e.g., PZT (Goser et al., 1990), etc. The main problems with these techniques arise from process-parameter variations across the chip, limited storage times (volatility), and lack of compatibility with standard VLSI processing technology. The weight sets for these trainable chips are obtained by training on a remote system (PC or workstation) and are then downloaded onto the chip. Then another short learning phase can be carried out in the chip used for the forward phase, and the remote system updates the weights until the network stabilizes. This yields a weight matrix that is adjusted to compensate for the inevitable disparities in analog computations due to process variance. This 'chip in loop' method has been used for Intel's analog ETANN chip (Tam et al., 1990), and for the Epsilon chip (Churcher et al., 1992). It should be clear that these chips are suited for many different applications, but do not allow for on-board training.

In order to get the benefits of fast analog implementation and the adaptivity properties of neural networks, one has to implement learning mechanisms on the chip. Only then can the adaptive real-time aspects of neural networks be fully exploited. However, the implementation of most learning rules into analog VLSI turns out to be very hard. One of the problems in multi-layered networks is that the target values of the hidden nodes are not defined. The backpropagation method gets around this by passing error signals recursively backwards from the output layer, estimating the effect of intermediate weight changes on each error signal via a relatively complex backwards pass. Information is non-local, which renders extra difficulties for implementation. In order to overcome these difficulties many research groups are investigating learning methods that better suit implementation in analog circuits. The development of these new rules is

also constrained by the implementation demands. Most proposed methods use a kind of weight perturbation which only requires a feed forward phase. These methods have proved to be quite successful, see for instance: weight perturbation (Jabri, 1991; Cauwenbergs, 1994), simultaneous perturbation (Maeda, 1993), random weight changing (Hirotsu, 1993; Murasaki, 1993), and virtual target strategy (Woodburn, 1994).

Although analog chips will never reach the flexibility attainable with digital chips, their speed and compactness make them very attractive for neural network research, especially when they adopt the adaptive properties of the original neural network paradigms. A final promising advantage is that they more directly interface with the real, analog world, whereas digital implementations will always require fast analog-to-digital converters to read in world information and digital-to-analog converters to put their data back into the world. We will now list some purely analog programmable and adaptive neuro chips.

ETANN (Intel Inc., CA, USA)
This Electrically Trainable Analog Neural Network (801770NX) (Holler et al., 1989; data sheet Intel, 1992) was the first commercially available analog neurochip which supports downloading of trained weights, but no on-board learning. The analog weights are stored as the voltage difference between two floating gates. The chip contains 64 neurons and 10,240 synapses, and performs at about 2GCPS. See also: Holler et al. (1989), Garth (1987), Graf et al. (1988), and Lindsey et al. (1992). The Mod2 neurocomputer (see below) is based on this chip.

The Mod2 Neurocomputer (Naval Air Warfare Center Weapons Division, CA)
This system incorporates neural networks as subsystems in a layered hierarchical structure. The Mod2 is designed to support parallel processing of image data at sensor (real-time) rates. The architecture was inspired by the structures of biological olfactory, auditory, and visual systems. The basic structure is a hierarchy of locally densely connected, globally sparsely connected networks. The locally densely interconnected network is implemented in a modular/block structure based upon the ETANN chip (see above). Mod2 is said to implement several neural network paradigms, and is in theory infinitely extensible (Mumford, 1992). An initial implementation (1992) consists of 12 ETANN chips, each able to perform 1.2 GCPS.

Fully analog chip (Kansai University, Japan)
Maeda et al. (1993) implemented a fully analog chip with a learning rule that is based on simultaneous perturbation. This does not need a backward phase as in the backpropagation rule, which simplifies implementation. The prototype chip consist of a neuron unit (which implements one neuron), a learning unit, and a control unit. It was shown that the chip could learn the XOR problem and the TCLX problem, where nine input cells and the combination of two output cells represent the letters T, C, L, and X.

Other fully analog approaches are described by: Li et al. (1993), Rao et al. (1990), Mueller et al. (1989), Fisher et al. (1991), Graf et al. (1988), Chambost and Sourier (1989), Murasaki et al. (1993), Hirotsu and Brooke (1993), Van der Spiegel et al. (1994), and Withagen (1994).

4.3 Neurocomputers built from hybrid neurochips

Both digital and analog techniques offer unique advantages, as was discussed in the former sections but they also have drawbacks with regard to their suitability for neural network implementations. The main shortcomings of digital techniques are the relative slowness of computation and the large amount of silicon and power that is required for multiplication circuits. Shortcomings of analog techniques are, for instance, the sensitivity to noise and susceptibility to interference and process variations. The right mixture of analog and digital techniques for the implementation of these processes will be very advantageous. In order to gain advantages of both techniques, and avoid the major drawbacks, several research groups are implementing hybrid systems. We distinguish normal hybrid implementations and pulse coded implementations.

ANNA (AT&T Bell Labs, NJ, USA)
This (Analog Neural Network Arithmetic and logic unit) chip can be used for a wide variety of neural network architectures (see Säckinger et al., 1992 for an OCR application) but is optimized for locally connected, weight-sharing networks and time-delay neural networks (TDNNs). Synaptic weights are trained off chip, quantized to the chip's resolution, and then downloaded into the chip's weight memory. They are represented by voltages. The interface to the chip is purely digital with two on-chip DACs converting the 6-bit digital weight values into the appropriate voltages. The system board for the ANNA chip is provided by a floating point DSP-32C for the learning process and calculation of the output layer of the backpropagation network. The ANNA chip comprises 4096 synapses and 8 linear neurons, and can handle up to 256 neural state inputs (Baxter, 1992). Performance: 5000 MCPS (peak), 1000 to 2000 MCPS (average).

Examples of other normal hybrid designs are reported in DeYong et al. (1992), Passos Almeida and Franca (1993), and Peiris et al. (1994). Pulse coding techniques were introduced by A.F. Murray (e.g., Hamilton et al., 1992; Murray, 1992; Murray & Smith, 1987; Murray et al., 1991; Murray & Tarassenko, 1993). They form a very promising approach to hybrid neural system building. In pulse coding techniques, the analog neural states are represented as sequences of pulses. This offers a number of advantages with regard to power consumption, calculations and their propagation. For instance, when using Pulse Frequency Modulation (PFM) the analog value is represented in the pulse density of the signal. In other words, the probability that a pulsed signal is logical 1 represents the 'analog' value for the signal. Addition of two signals can then easily be done by logical OR-ing the signals because this will add the probabilities. Multiplication in its turn can be done by logical AND-ing the signals. In this way the fundamental computations can be built from simple logical gates. These techniques have been investigated recently and many implementational solutions have been proposed (see for instance Han et al., 1993; Nijhuis, 1992; Oteki et al., 1993; Shin, 1993; Tomberg et al., 1992; Tomlinson & Walker, 1990; Verleysen & Jespers, 1991). The applicability to neural network implementation follows from several studies regarding accuracy and precision. Compared to straightforward digital techniques, where one can define the resolution of values as precisely as one wants (just by adding more bits), these pulse stream techniques are not that precise. Studies by Murray et al. (1991) and Siggelkow (1991) show that this does not necessarily have to restrain their applicability to the implementation of neural networks. The implemented neural networks were robust enough for errors within certain bounds. Here again we see a reverse engineering interplay between paradigms and implementation which could yield strong paradigms suitable for hardware implementation. The use of pulses is inspired by looking at real neurons which, themselves interact with pulse trains. In short, two main benefits when using pulse coding can be seen: (1) analog circuitry (e.g., multipliers) occupies a much smaller silicon area than the digital equivalent, and (2) analog values encoded in digital states are robust and easily transmitted within chips or across chip boundaries.

Epsilon (Dept. of Electrical Engineering, Edinburgh University, UK)
The (Edinburgh Pulse Stream Implementation of a Learning Oriented Network) is a large-scale generic building block device (Baxter, 1992). It consists of 30 nodes and 3600 synaptic weights, and can be used both as a 'save' accelerator to a conventional computer and as an 'autonomous' processor. The chip has a single layer of weights but can be cascaded to form larger networks. The synapses are formed by transconductance multiplier circuits which generate output currents proportional to the product of two input voltages. A weight is represented by fixing one of these voltages. Both synchronous and asynchronous neuron designs are included in the Epsilon chip. The first uses pulse width modulation and is specially designed with vision applications in mind. The second uses pulse frequency modulation, which is advantageous for feedback and recurrent networks, where temporal characteristics are important. The synchronous implementation was successfully applied to a vowel recognition task. An MLP network consisting of 38 nodes (hidden and output) was trained by the 'chip in loop method' and showed performance comparable to a software simulation on a SPARC station. With this chip it has been shown that it is possible to implement robust and reliable networks using the pulse stream technique. Performance: 360 MCPS (Baxter, 1992; Churcher et al., 1992).

Other pulse coding implementations are described in Han et al. (1993), Nijhuis (1992), Oteki et al. (1993), Schwartz (1990), Shin (1993), Tomberg et al. (1992), Tomlinson and Walker (1990), and Verleysen (1992).

5 Discussion

This paper has presented an overview of neural hardware projects. The architectures have been classified into the following main categories: accelerator boards, multiprocessor systems, and digital, analog, and hybrid neurochips. The successive order of these categories corresponds to increasing speed-up but decreasing maturity of the techniques used. Accelerator boards have been developed to completion and are commercially available, whereas hybrid neurochips can thus far only be found in research laboratories. Digital techniques offer the best possibilities for implementing flexible, general purpose neurocomputers. Analog techniques offer great advantages concerning compactness and speed, but much research still has to be done in order to build large-scale, adaptive systems from them. Combining the best of analog and digital techniques would rule out implementation-specific problems. Although still in their infancy, these hybrid approaches are promising. Implementation projects such as those reported above are giving rise to new insights in defining learning paradigms, insights that most likely should have never emerged from simulation studies.

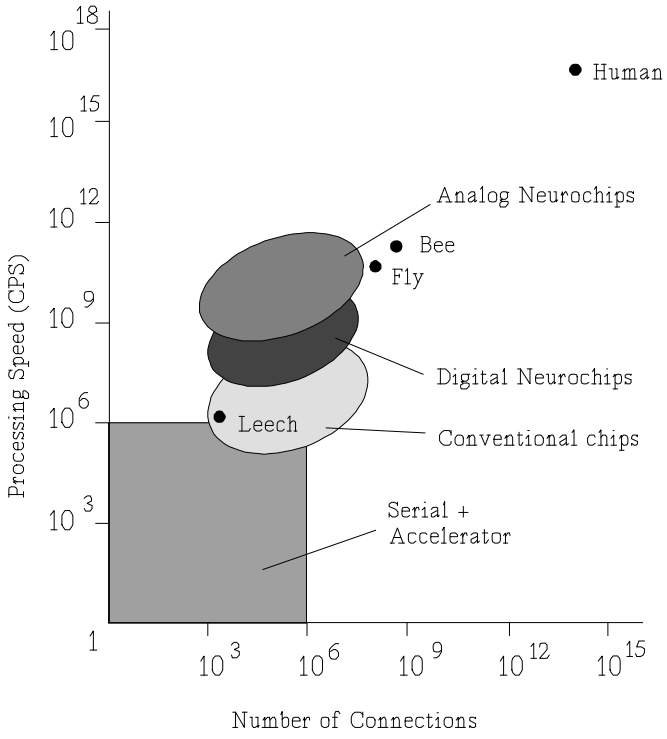

Fig. 2. Neurocomputer performances (after Glessner and Pöchmüller, 1994).

Figure 2 presents the performances of several architectures. The size of a neurocomputer is expressed in the number of connections. The speed is measured in CPS. The positions of the neurocomputers in the chart are derived from the publications and serve here only as an indication. The serial implementations are all clustered in the bottom-left corner. Massively parallel multiprocessor systems based on analog neurochips

will deliver the most speed-up and largest size. For comparison, the chart also shows four biological implementations. The performance of the human brain, expressed in numbers of connections and numbers of connections per second, far outranges artificial implementations. Special neurochips mainly increase the number of connections to be processed per second. In order to scale-up the size (i.e., number of nodes and connections) large numbers of processing elements must be connected in massively parallel architectures.

In large digital multiprocessor designs there is a common tendency regarding the communication strategy. Many architectures use a localized communication plan in contrast to the full connectivity found in analog designs. In multiprocessor architectures this is accomplished by combining neurochips with communication chips. Three architectural classes can be distinguished: systolic ring architectures, broadcast bus architectures, and hierarchical architectures. Systolic arrays are considered non-scalable. Broadcasting is according to many designers (e.g., Hammerstrom, 1990; Hecht-Nielsen, 1990) the most efficient multiplexed interconnection architecture for large fan-in and fan-out. It seems that broadcast communication is often the key to success in getting communication and processing balanced, since it is a way to time-share communication paths efficiently. Electronic broadcast structures, however, have the limitation that the number of processing resources must be less than the fan-out capability of a single bus line driver. This problem can be overcome by implementing several local broadcast structures in a hierarchical way.

Current problems in large-scale systems are posed by downloading and initializing the systems, and pre- and post-processing for non-connectionist applications, e.g., normalizing inputs, chunking matrices into vectors, AD-conversion (Baxter, 1992; Heemskerk et al., 1994). To implement these kinds of operations efficiently, large-scale neurocomputers should also contain conventional algorithmic units. Much research has to be devoted to the integration of neural and conventional units. A combination of neural and rule based hardware with sufficient I/O possibilities, and the support of on-line learning, will lead to action oriented systems. Another issue that was not addressed by this paper, but needs to be considered when dealing with neurocomputers, is the integration of software models and large-scale hardware. Hardware will only be successful when accompanied by dedicated software. Software environments for neurocomputers have to support programmability and ease of use (Means, 1994; Murre and Kleynenberg, 1991).

Compact implementations of neural computers lead to a neuromorphic approach where artificial (silicon) protheses replace their biological equivalents (Rückert, 1993). This novel methodology was inspired by the work of Carver Mead (Mead & Mahowald, 1989), which led to interesting prototype chips that are used as a silicon retina, and an electronic cochlea (Lazzaro & Mead, 1989). A brain-like digital implementation related to this neuromorphic approach and reported in this paper was proposed by Richert et al. (1993). The flexibility of their design with regard to the implementation of local rules opens a testbed for the development of new and even more plausible paradigms (see for instance Eckmiller & Napp-Zinn, 1993).

In the next decade micro-electronics will most likely continue to dominate the field of neural network implementation. If progress advances as rapidly as it has in the past, this implies that neurocomputer performances will increase by about two orders of magnitude. Consequently, neurocomputers will be approaching TeraCPS (10^{12} CPS) performance. Networks consisting of 1 million nodes, each with about 1,000 inputs, can be computed at brain speed (100-1000 Hz). This would offer good opportunities to experiment with reasonably large networks.

References

Adaptive Solutions (1993). *CNAPS Server/64.* Cromenco S.A. Benelux N.V., various data sheets Adaptive Solutions Inc., Beaverton, Oregon,

Agranat, A.J., C.F. Neugebauer, & A. Yariv (1990). A CCD Based Neural Network Integrated Circuit with 64K Analog Programmable Synapses. *Proceedings of the IJCNN-90-San Diego CA*, II, 151-555.

Aleksander, I., W. Thomas, & P. Bowden (1984). "WISARD, a radical new step forward in image recognition". *Sensor review*, 120-124.

Aleksander, I. (1990). Ideal Neurons for Neural Computers. *Proceedings of the International Conference on Parallel Processing in Neural Systems and Computers*, Düsseldorf, Springer Verlag.

Arbib, M.A., (1989). Schemas and neural networks for sixth generation computing. *Journal of Parallel Distributed Computing*, 6, 2, 185-216.

Arif, A.F., S. Kuno, A. Iwata, Y. Yoshida (1993). A Neural Network Accelerator Using Matrix Memory with Broadcast Bus. *Proceedings of the IJCNN-93-Nagoya*, 3050-3053.

Asanovic, K., J. Beck, J. Feldman, N. Morgan, & J. Wawrzynek (1993). Development of a Connectionist Network Supercomputer. *Proceedings of the 3rd International Conference on Microelectronics for Neural Networks (Micro Neuro)*, April 1993, Edinburgh, 253-262.

Bakkers, A.P. (1988). Neural controllers and Transputers. Presented at the BIRA seminar, Antwerp Belgium.

Barber, S.M., J.G. Delgado-Frias, S. Vassiliadis, & G.G. Pechanek (1993). SPIN-L: Sequential pipelined neuroemulator with learning capabilities. *Proceedings of the IJCNN-93-Nagoya*, 1927-1930.

Baxter, D.J. (1992). Process-Tolerant VLSI Neural Networks for Applications in Optimisation. PhD thesis, The University of Edinburgh.

Beichter, J., N. Brüls, E. Sicheneder, U. Ramacher, & H. Klar (1993). Design of a general-purpose neural signal processor. *Neurocomputing*, 5, 17-23.

Bengtsson, L., A. Linde, B. Svensson, M. Tavenkiku, A. Ahlander (1993). The REMAP Massively Parallel Computer for Neural Computations. *Proceedings of the 3rd International Conference on Microelectronics for Neural Networks (Micro Neuro)*, Edinburgh, 47-62.

Carling, A. (1992). *Introducing Neural Networks*. Sigma Press, Wilmslow, UK.

Castillo, F., J. Cabestany, & J.M. Moreno (1992). The Dynamic Ring Architecture. In: I. Aleksander, & J. Taylor (Eds.). *Artificial Neural Networks II*, Proceedings of the ICANN-92-Brighton UK, Amsterdam: Elsevier, 1439-1442.

Cauwenberghs, G. (1994). An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory, Submitted to IEEE Transactions on Neural Networks.

Chambost, E. De, & M. Sourier (1989). Implementing classifications on one bit connection analog programmable neural network. *Proceedings of the IJCNN-89-Washington DC.*

Chinn, G., K.A. Grasjki, C. Chen, C. Kuszmaul, & S. Tomboulian (1990). Systolic array implementation of neural nets on the MasPar MP-1 massively parallel processor. *Proceedings of the IJCNN-90-San Diego*, CA, II-169-173.

Churcher, S., D.J. Baxter, A. Hamilton, A.F. Murray, & H.M. Reekie (1992). Generic analog neural computation - the Epsilon chip. Advances in Neural Information Processing Systems: *Proceedings of the 1992 Conference*, Denver Colorado.

DeYong, M.R., R.L. Findley, & C. Fields (1992). The Design, Fabrication, and Test of a New VLSI Hybrid Analog-Digital Neural Processing Element. *IEEE Transaction on Neural Networks*, 3, 3, 363-374.

Duranton, M. & J.A. Sirat (1989). Learning on VLSI: A general purpose digital neurochip. *Proceedings of the IJCNN-89-Washington DC.*

Eckmiller, R., & H. Napp-Zinn (1993). Information Processing in Biology-Inspired Pulse Coded Neural Networks. *Proceedings of the IJCNN-93-Nagoya*, 643-648.

Eide, Å., Th. Lindblad, C.S. Lindsey, M. Minerskjöld, G. Sekhniaidze, & G. Székely (1994). An implementation of the Zero Instruction Set Computer (ZISC036) on a PC/ISA-bus card. *Proceedings of the WNN/FNN-94-Washington DC*, (ftp://archive.cis.ohio-state.edu/pub/neuroprose)

Ernst, H.P. & B. Mokry, & Z. Schreter (1990). A transputer based general simulator for connectionist models. In: R. Eckmiller, G. Hartmann, & G. Hasuke (Eds.). Parallel processing in neural systems and computers. Amsterdam: North-Holland, 283-286.

Fisher W.A., M. Okamura, R.C. Smithson, D. Specht (1991). *Implementation of the Hopfield decomposition and DFT into Analog Hardware*. Technical Report 0/91-10 B/256 Lockheed Palo Alto Research Laboratory.

Foo, S.K., P. Saratchandran, N. Sundararajan (1993). Parallel Implementation of Backpropagation on Transputers. *Proceedings of the IJCNN-93-Nagoya*, 3058-3061.

Fujimoto, Y., N. Fukuda, & T. Akabane (1992). Massively parallel architectures for large scale neural networks simulations. *IEEE Transactions on Neural Networks*, 3, 6, 876-888.

Garth, S.C.J. (1987). A Chipset for High Speed Simulation of Neural Network Systems. *Proceedings of the IEEE first Int. Conf. on Neural Networks*, III-443-452.

Glesner, M., & M. Pöchmüller (1994). Neurocomputers. An overview of neural networks in VLSI. Chapman & Hall, Neural Computing series, London.

Goser, K., U. Ramacher, & U. Rückert (1990). Microelectronics for Neural Networks. Proceedings of the 1st Int. Workshop, University of Dortmund.

Grachev, L.V., & E.J. Kirsanov (1993). The general purpose neurocomputer "Hercules". *Proceedings of the IJCNN-93-Nagoya*, 1935-1938.

Graf, H.P., L.D. Jackel & W. E. Hubbard (1988). VLSI Implementation of a Neural Network Model. *Computer*, March, 41-49.

Grasjki, K.A. (1992). Neurocomputing using the MasPar MP-1. In: Przytula and Prasanna (Eds.). *Digital Parallel Implementations of Neural Networks.* Prentice Hall, Englewood Cliffs, NJ.

Hamilton, A., A.F. Murray, D.J. Baxter, S. Churcher, H.M. Reekie, & L. Tarassenko (1992). Integrated Pulse Stream Neural Networks: Results, Issues, and Pointers. *IEEE Transaction on Neural Networks*, 3, 3, 385-393.

Hammerstrom, D. (1990). A VLSI Architecture for High Performance, Low-Cost, On-chip Learning. *Proceedings of the INNC-90-Paris*, 2, 537-543.

Hammerstrom, D., & E. Means (1990). System Design for a Second Generation Neurocomputer. *Proceedings of the IJCNN-90-Washington DC*, II-80-83.

Hammerstrom, D., & N. Nguyen (1991). An Implementation of Kohonen's Self-Organizing Map on the Adaptive Solutions Neurocomputer. In: T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Eds.). *Artificial Neural Networks.* Proceedings of the ICANN-91-Helsinki, Amsterdam: Elsevier, 715-720.

Han, I., K. Ahn, H. Park (1993). Electronic Implementation of Artificial Neural Network with URAN. *Proceedings of the IJCNN-93-Nagoya*, 1963-1966.

Hanson, W.A., C.A. Cruz., & J.Y. Tam (1987). CONE - Computational Network Environment. *Proc. IEEE First Int. Conf. on Neural Networks*, III-531-538.

Hecht-Nielsen, R. (1990). *Neurocomputing*. Addison-Wesley Publishing Company.

Heemskerk, J.N.H. (1995). *Neurocomputers for Brain-Style Processing. Design, Implementation and Application*. PhD thesis, Unit of Experimental and Theoretical Psychology Leiden University, The Netherlands.

Heemskerk, J.N.H., J. Hoekstra, J.M.J. Murre, L.H.J.K. Kemna, & P.T.W. Hudson (1994). The BSP400: a modular neurocomputer. *Microprocessors and Microsystems*, 18, 2, 67-78.

Heemskerk, J.N.H., J.M.J. Murre, J. Hoekstra, L.H.J.G. Kemna & P.T.W. Hudson (1991). The BSP400: a modular neurocomputer assembled from 400 low-cost microprocessors. In: T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Eds.). *Artificial Neural Networks.* Proceedings of the ICANN-91-Helsinki, Amsterdam: Elsevier, 709-714.

Heemskerk, J.N.H., & J.M.J. Murre (1995). *Brain-size neurocomputers: analyses and simulations of neural topologies mapped on Fractal Architectures*. Dept. of Experimental and Theoretical Psychology, Leiden University, The Netherlands, submitted.

Heemskerk, J.N.H., J.M.J. Murre, A. Melissant, M. Pelgrom, & P.T.W. Hudson (1992). MindShape a neurocomputer concept based on the fractal architecture. In: I. Aleksander I, & J. Taylor (Eds.). *Artificial Neural Networks II*: Proceedings of the ICANN-92-Brighton UK. Amsterdam: Elsevier, 1483-1486.

Hiraiwa, A., S. Kurosu, S. Arisawa & M. Inoue (1990). A two level pipeline RISC processor array for ANN. *Proceedings of the IJCNN-90-Washington DC*, II-137-140.

Hirotsu, K. & M.A. Brooke (1993). An Analog Neural Network Chip with Random Weight Change Learning Algorithm. *Proceedings of the IJCNN-93-Nagoya*, 3031-3034.

Hitachi (1993). *My-Neuropower*. datasheet. 5-22-1 Jousuihon-cho, Kodaira-shi, Tokyo.

HNC (1993). High-Performance Parallel Computing. *Simd Numerical Array Processor*. Data Sheet. 5501 Oberlin Drive, San Diego, CA, USA.

Hoekstra, J. (1990). An integrated artificial neuron based on junction-ccd technology. *Proceedings of the INNC-90-Paris*, 559-562.

Holler, M., S. Tam, H. Castro, R. Benson (1989). An Electrically Trainable Artificial Neural Network (ETANN) with 10240 "Floating Gate Synapses". *Proceedings of the IJCNN-89-Washington DC*, II-191-196.

Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences USA*, 79, 2554-2558.

Hudson, P.T.W., J. Hoekstra & J.N.H. Heemskerk (1991). CALM Modules on a 400-processor machine. *Proceedings of the SNN Symposium on Neural Networks*, 33-35.

Ienne, P. (1993a). Quantitative Comparison of Architectures for Digital Neuro-computers. *Proceedings of the IJCNN-93-Nagoya*, 1987-1990.

Ienne, P. (1993b). Architectures for Neuro-Computers: Review and Performance Evaluation. Technical Report no. 93/21. Swiss Federal Institute of Technology, Lausanne.

Ienne, P., & M.A. Viredaz (1993). GENES IV: A Bit-Serial Processing Element for a Multi-Model Neural Network Accelerator. In: L. Dadda & B. Wah (Eds.). Proceedings of the International Conference on Application-Specific Array Processors, 345-356, Venezia.

Intel (1992). *Neural Network Technology & Applications*. Data Sheet and several articles.

Iwata, A., Y. Yoshida, S. Matsuda, Y. Sato, & N. Suzumura (1989). An Artificial Neural Network Accelerator using General Purpose 24 Bits Floating Point Digital Signal Processors. *Proceedings of the IJCNN-89-Washington DC*, II-171-175.

Jabri, M. & B. Flower (1991). Weight Perturbation: An optimal architecture and learning technique for analog VSLI feedforward and recurrent multilayer networks. *Neural Computation*, 3, 546-565.

Jackel, L.D., H.P. Graf., & R.E. Howard (1987). Electronic Neural Network chips. *Applied Optics*, 26, 5077-5080.

Kato, H, H. Yoshizawa, H. Iciki & K. Asakawa (1990). A Parallel Neurocomputer Architecture towards Billion Connection Updates Per Second. *Proceedings of the IJCNN-90-Washington DC*, 2, 47-50.

Kohonen, T. (1989). *Selforganization and associative memory*, 3rd edition, Berlin Germany: Springer-Verlag.

Kotilainen, P., J. Saarinen, & K. Kaski (1993a). Neural network computation in a parallel multiprocessor architecture. *Proceedings of the IJCNN-93-Nagoya*, 1979-1982.

Kotilainen, P., J. Saarinen & K. Kaski (1993b). A multiprocessor architecture for general purpose neurocomputing applications. *Micro neuro. Proceedings of the third International Conference on Microelectronics for Neural Networks*, 6-8 April 1993. 35-45.

Landsverk, O., J. Greipsland, J.G. Solheim, J. A. Mathisen, H. Dahle, & L. Utne (1992). RENNS - a Reconfigurable Neural Network Server. In: I. Aleksander & J. Taylor (Eds.). *Artificial Neural Networks II*, Proceedings of the ICANN-92, Brighton UK, Amsterdam: Elsevier, 1455-1459.

Lazzaro, J. & C.A. Mead (1989). A Silicon Model of Auditory Localization. *Neural Computation*, 1, 47-57.

Leung C.S., & R. Setiono (1993). Efficient neural network training algorithm for the Cray Y-MP supercomputer. *Proceedings of the IJCNN-93-Nagoya*, 1943-1946.

Li, B., Z. Li, & B. Shi (1993). An Analog Integrated Circuit of A Hamming Neural Network Designed And Fabricated in CMOS Technology. *Proceedings of the IJCNN-93-Nagoya*, 879-882.

Lindsey, C., B. Denby, & H. Haggerty (1992). Real-time track finding in a drift Chamber with a VLSI Neural Network. In: Intel (1992).

Lippmann, R.P. (1991). A Critical Overview of Neural Network Pattern Classifiers. *Proceedings of the 1991 Workshop on Neural Networks for Signal Processing*, 266-275.

Maeda, Y., H. Hirano, & Y. Kanata (1993). An Analog Neural Network Circuit with a Learning Rule via Simultaneous Perturbation. *Proceedings of the IJCNN-93-Nagoya*, 853-856.

Mauduit, N., M. Duranton, J. Gobert, J. Sirat (1991). Building Up Neuromimetic Machines with LNeuro 1.0. *Proceedings of the IJCNN-91-Singapore*, 602-607.

Mauduit, N., M. Duranton, J. Gobert, & J.A. Sirat (1992). Lneuro 1.0: A Piece of Hardware LEGO for Building Neural Network Systems. *IEEE Transactions on Neural Networks*, 3, 3, 414-422.

Mead, C.A. & M.A. Mahowald (1988). A silicon model of early visual processing. *Neural Networks*, 1, 91-97.

Means, R. (1994). High speed parallel performance issues for neural network applications. *Proceedings of the IEEE ICNN-94-Orlando Florida*, 10-16.

Micro Devices (1990). Neural Bit Slice, Data Sheet. *Neural Circuits for Todays Applications.* 30 Skyline Drive, Lake Mary, FL, USA.

Migliore, M., G.F. Ayala, & S.L. Fornili, S.L., (1990). Modelling of Neuronal Systems on Transputer Networks. In: G. Hauske (Ed.). *Parallel Processing in Neural Systems and Computers*, Elsevier Science Publishers B.V. (North Holland), 291-294.

Moody, R.E., C.J. Darken (1989). Fast Learning in networks of locally tuned processing units. *Neural Computation*, 1, 281-294.

Morgan, N. (1993). Making Useful Neurocomputers. *Proceedings of the 3rd International Conference on Microelectronics for Neural Networks (Micro Neuro)*, April 1993, Edinburgh, 297-305.

Morgan, N., J. Beck, P. Kohn, J. Bilmes, E. Allman, & J. Beer (1992). The Ring Array Processor: A Multiprocessing Peripheral for Connectionist Applications. *Journal of Parallel and Distributed Computing*, 14, 248-259.

Mueller, P., J. van der Spiegel, D. Blackman, T. Chiu, T. Clare, J. Dao, C. Donham, T. Hsieh & M. Loinaz (1989). A general purpose analog neural computer. *Proceedings of the IJNNC-89-Washington DC*, 2, 177-182.

Mumford, M.L., D.K. Andes, & L.R. Kern (1992). The Mod2 Neurocomputer System Design. IEEE Transactions on Neural Networks, 3, 3, 423-433.

Murasaki, M., Y. Arima, & H. Shinohara (1993). A 20-Tera CPS Analog Neural Network Board. *Proceedings of the IJCNN-93-Nagoya*, 3027-3030.

Murray, A.F. (1992). Analog Neural VLSI: Issues, Trends and Pulses. In: I. Aleksander, & J. Taylor (Eds.). *Artificial Neural Networks II.* Proceedings of the ICANN-92-Brighton UK, Amsterdam: Elsevier, 35-44.

Murray, A.F., & A.V.W. Smith (1987). Asynchronous arithmetic for VLSI neural systems. *Electron. Lett.*, 23, 12, 642-643.

Murray, A.F., D. Del Corso, & L. Tarassenko (1991). Pulse-Stream VLSI Neural Networks Mixing Analog and Digital Techniques. *IEEE Transactions on Neural Networks*, 2, 2, 193-204.

Murray, A.F., & L. Tarassenko (1993). Analogue Neural VLSI, A pulse stream approach. Chapman & Hall, Neural Computing Series 2, London.

Murre, J.M.J. (1995). Neurosimulators. To appear in: M.A. Arbib (Ed.). *Handbook of Brain Research and Neural Networks*, MIT Press.

Murre, J.M.J. & S.E. Kleynenberg (1991). Extending the MetaNet Network Environment: process control and machine independence. In: T. Kohonen, K. Mäkisara, O. Simula, J. Kangas (Eds.). *Artificial Neural Networks.* Proceedings of the ICANN-91-Helsinki. Amsterdam: Elsevier, 545-550.

Neural Technologies Limited (1993). *A company profile*. 7a Lavant Street Petersfield, Hampshire GU32 3EL, UK.

Nijhuis, J.A.G. (1992). *An Engineering Approach to Neural System Design*. PhD Thesis.

Nordström, T., & B. Svensson (1992). Using and Designing Massively Parallel Computers for Artificial Neural Networks. *Journal of Parallel and Distributed Computers*, 14, 260-285.

Ntourntoufis, P. (1992). Parallel Hardware Implementation of the General Neural Unit Model. In: I. Aleksander, & J. Taylor (Eds.). *Artificial Neural Networks II*, Proceedings of the ICANN-92-Brighton UK, Amsterdam: Elsevier, 1443-1446.

Onuki, J., T. Maenosono, M. Shibata, N. Iijima, H. Mitsui, Y. Yoshida, & M. Sobne (1993). ANN Accelerator by Parallel Processor Based on DSP. *Proceedings of the IJCNN-93-Nagoya*, 1913-1916.

Oteki, S., A. Hashimoto, T. Furuta, & S. Motomura (1993). A Digital Neural Network VLSI with On-chip Learning Using Stochastic Pulse Encoding. *Proceedings of the IJCNN-93-Nagoya*, 3039-3045.

Ouali, J., & G. Saucier (1990). Fast Generation of Neuro-Asics. *Proceedings of the IJCNN-90-San Diego*, 2, 557-561.

Ouali, J., G. Saucier, & Trilhe (1991). Fast design of digital dedicated neurochips. In: U. Ramacher & U. Rückert (Eds). *VLSI design of Neural Networks*. Kluwer academic publishers.

Panda, D.K. & K. Hwang (1990). A Multiple-Bus Network for Implementing Very-Large Neural Networks with Back-Propagation Learning. In: B. Widrow, B. Angeniol (Eds.). *Proceedings of the INNC-90-Paris*, 2, 175-178.

Passos Almeida, A., & J.E. Franca (1993). A mixed-mode architecture for implementation of analog neural networks with digital programmability. *Proceedings of the IJCNN-93-Nagoya*, 887-890.

Peiris, V., B. Hochet, & M. DeClercq (1994). Implementation of a fully parallel kohonen map: A mixed analog digital approach. *Proceedings of the IEEE ICNN-94-Orlando Florida*, 2064-2068.

Pomerlau, D.A., G.L. Gushoirla, D.S. Touretzky, & H.T. Kung (1988). Neural network simulation at Warp speed: how we got 17 million connections per second. *Proceedings of the IEEE ICNN-88*, 2, 143-150.

Ramacher, U. (1991). Synapse-X: A General-Purpose Neurocomputer Architecture. *Proceedings of the IJCNN-91-Singapore*, 2168-2176.

Ramacher, U. & U. Rückert (1991). *VLSI design of Neural Networks*. Kluwer academic publishers.

Ramacher, U., W. Raab, J. Anlauf, U. Hachmann, J. Beichter, N. Brüls, M. Weßeling, & E. Sicheneder (1993). Multiprocessor and Memory Architecture of the Neurocomputer SYNAPSE-1. *Proceedings of the 3rd International Conference on Microelectronics for Neural Networks (Micro Neuro)*, April 1993, Edinburgh, 227-231.

Rao, A., M.R. Walker, L.T. Clark, L.A. Akers & R.O. Grondin (1990). VLSI Implementation of Neural Classifiers. *Neural Computation*, 2, 35-43.

Richert, R., B.J. Hosticka, M. Kesper, M. Scholles, & M. Schwarz (1993). An Emulator for Biologically-Inspired Neural Networks. *Proceedings of the IJCNN-93-Nagoya*, 841-844.

Rumelhart, D.E. & J.L. McCelland (Eds.) (1986). *Parallel Distributed Processing. Volume I: Foundations*. Cambridge MA: MIT Press.

Rückert, U. (1993). Microelectronic Implementation of Neural Networks. In: H. Hüning, S. Neuhauser, M. Raus, & W. Ritschel (Eds.). *Workshop on Neural Networks at RWTH Aachen*, July 1993, 77-86.

Säckinger, E., B.E. Boser, J. Bromley, Y. LeCun, & L.D. Jackel (1992). Application of the ANNA Neural Network Chip to High Speed Character Recognition. *IEEE Transactions on Neural Networks*, 3, 3, 498-505.

Sato, Y, K. Shibata, M. Asai, M. Ohki, M. Sugie (1993). Development of a high-performance general purpose neuro-computer composed of 512 digital neurons. *Proceedings of the IJCNN-93-Nagoya*, 1967-1970.

Savely, R.T. (1987). The implementation of Neural Network Technology. *Proceedings of the IEEE first Conference on artificial neural networks*, IV-447-484.

Schwartz, T.J., (1990). A Neural chips survey. *AI Expert*, 5, 12, 34-39.

Sejnowsky, T. & C. Rosenberg (1986). *NETtalk: A Parallel Network that learns to read aloud.* John Hopkins University Electrical Engineering and Computer Science Department, JHU/EECS-86/01.

Shams, S. & J. Gaudiot (1990). Massively parallel implementations of neural networks: a comperative analysis. *IEEE Transactions on Computer.*

Shams, S. & J. Gaudiot (1992). Efficient implementation of neural networks on the DREAM machine. Proceedings of the 11th International Conference on Pattern Recognition, The Hague, The Netherlands, 204-208.

Shin, J. (1993). Novel Impulse Neural Circuits for Pulsed Neural Network. *Proceedings of the IJCNN-93--Nagoya*, 3017-3022.

Siggelkow A., J. Nijhuis, S. Neußer, & B. Spaanenburg (1991). Influence of hardware characteristics on the performance of a neural system. In: T. Kohonen, K. Mäkisara, O. Simula, J. Kangas (Eds.). *Artificial Neural Networks.* Proceedings of the ICANN-91-Helsinki. Amsterdam: Elsevier, 697-702.

Singer, A (1990). Exploiting the inherent parallelism of artificial neural networks to achieve 1300 million interconnections per second. In: B. Angeniol & B. Widrow (Eds.). *Proceedings of INNC-90-Paris*, Dordrecht: Kluwer, 656-660.

Soegtrop, M., H.A. Klagges (1993). A Massively Parallel Neurocomputer. *Proceedings of the 3rd International Conference on Microelectronics for Neural Networks (Micro Neuro)*, April 1993, Edinburgh, pp 241-251.

Speckman, H., P. Thole, & W. Rosenstiel (1993). COKOS: A COprocessor for Kohonen's Selforganizing map. In: C.C.A.M. Gielen & B. Kappen (Eds.). *Proceedings of the ICANN-93-Amsterdam*, London: Springer-Verlag, 1040-1045.

Tam, S., B. Gupta, H. Castro, & M. Holler (1990). Learning on an Analog VLSI Neural Network Chip. *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics.*

Theeten, J.B., M. Duranton, N. Mauduit & J.A. Sirat (1990). The Lneuro-chip: a digital VLSI with on-chip learning mechanism. In: B. Angeniol & B. Widrow (Eds.). *Proceedings of the INNC-90-Paris.* Dordrecht: Kluwer, 593-596.

Tomberg, J., & K. Kaski (1992). VLSI Architecture of the self-organizing neural network using synchronous pulse-density modulation technique. In: I. Aleksander & J. Taylor (Eds.). *Artificial Neural Networks II*. Proceedings of the ICANN-92, Brighton UK, Amsterdam: Elsevier, 1431-1434.

Tomlinson, M.S. & D.J. Walker (1990). DNNA: A digital neural network architecture. In: B. Angeniol & B. Widrow (Eds.). *Proceedings of the INNC-90-Paris.* Dordrecht: Kluwer, 589-592.

Treleaven, P.C. (1989). Neurocomputers. *International Journal of Neurocomputing*, 1, 4-31.

Van der Spiegel, J., C. Donham, R. Etienne-Cummings, S. Fernando, & P. Mueller (1994). Large scale analog neural computer with programmable architecture and programmable time constants for temporal pattern analysis. *Proceedings of the IEEE ICNN-94-Orlando Florida*, 1830-1835.

Van Keulen, E., S. Colak, H. Withagen, & H. Hegt (1994). Neural Network Hardware Performance Criteria. *Proceedings of IEEE International Conference on Neural Networks*, June 28-July 2, 1885-1888.

Vellasco, M.M.B.R. (1992). A VLSI architecture for neural network chips. PhD thesis, Department of Computer Science, University of London.

Verleysen, M., & P. Jespers (1991). Stochastic computations in VLSI analog Neural Networks. In: T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Eds.). *Artificial Neural Networks.* Proceedings of the ICANN-91-Helsinki, Amsterdam: Elsevier, 691-696,

Viredaz, M.A. & P. Ienne (1993). MANTRA I: A Systolic Neuro-Computer. *Proceedings of the IJCNN-93-Nagoya*, 3054-3057.

Vuurpijl, L.G. (1992). Using Transputer Systems for Neural Network Simulations. *Proceedings of the SNN Symposium on Neural Networks*, 27-28.

Watanabe, T., Y. Sugiyama, T. Kondo, Y. Kitamura (1989). Neural Network Simulation on a Massively Parallel Cellular Array Processor: AAP-2. *Proceedings of the IJCNN-89-Washington DC*, II-155-161.

Witbrock, M. & M. Zagha (1989). *An Implementation of Back-Propagation learning on GF11, a Large SIMD Parallel Computer*. Internal Report CMU-CS-89-208, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Withagen, H. (1994). Implementing backpropagation with analog hardware. *Proceedings of the IEEE ICNN-94-Orlando Florida*, 2015-2017.

Woodburn, R., H.M. Reekie, & A.F. Murray (1994). Pulse-stream circuits for on-chip learning in analogue VLSI neural networks. *Proceedings of the IEEE International symposium on circuits and systems*, London, 103-106.

Yasunaga, M., N. Masuda, M. Yagyu, M. Asai, K. Shibata, M. Ooyama, & M. Yamada (1991). A Self-Learning Neural Network composed of 1152 Digital Neurons in Wafer-Scale LSIs. *Proceedings of the IJCNN-91-Singapore*, 1844-1849.