High Speed Neural Network Chip for Trigger Purposes in High Energy Physics

W. Eppler[1], T. Fischer[1], H. Gemmeke[1], A. Menchikov[2]

[1] Forschungszentrum Karlsruhe (FZK), POB 3640, 76021 Karlsruhe, Germany

[2] Joint Institute for Nuclear Research (JINR), 141980 Dubna, Russia

Abstract

A novel neural chip SAND (Simple Applicable Neural Device) is described. It is highly usable for hardware triggers in particle physics. The chip is optimized for a high input data rate (50 MHz, 16 bit data) at a very low cost basis. The performance of a single SAND chip is 200 MOPS due to four parallel 16 bit multipliers and 40 bit adders working in one clock cycle. The chip is able to implement feedforward neural networks with a maximum of 512 input neurons and three hidden layers. Kohonen feature maps and radial basis function networks may be also calculated. Four chips will be implemented on a PCI-board for simulation and on a VME board for trigger and on- and off-line analysis.

1. Introduction

Currently at FZK, in collaboration with IMS, the neurochip SAND (Simple Applicable Neural Device) is under development for on- and off-line data analysis as well as first and second level triggers in astrophysics experiments (KASCADE, MILAGRO, AUGER). Many sophisticated methods were proposed and implemented during the last decade to reveal the characteristic of extensive air showers. But one drawback is still present - the absence of an 'intelligent' adaptive hardware trigger and on-line data analysis. Usually it is a multiplicity or sum-energy trigger with very simple logic, requiring some channels exceeding the chosen threshold value. Now detailed simulations of an Extensive Air Shower (EAS) developing in the atmosphere and the response of the apparatus will be used to train an Artificial Neural Network (ANN). This allows one to implement sophisticated pattern recognition tasks for first level trigger and event builders in modern EAS experiments, like KASCADE in Karlsruhe, measuring as many parameters of a single event as possible. Information from thousands of electronic channels has to be processed in a very short time. The fast primary energy and primary particle type estimators will be trained by simulations and implemented for on-line analysis.

For most of these applications a feedforward network is used. In feedforward networks neurons are arranged in layers without back-loops. There are no connections between neurons of the same layer. The basic element of a neural network is an artificial neuron described by

$$x_i = f\left(\sum_{j=1}^{n} w_{ij} \cdot o_j + \Theta\right) \tag{1}$$

where w_{ij} are the connection weights from neurons j to neuron i. The input activities o_j are multiplied with the connection weights, accumulated and transfered by a nonlinear activation function to the output activities x_i (Fig. 1).

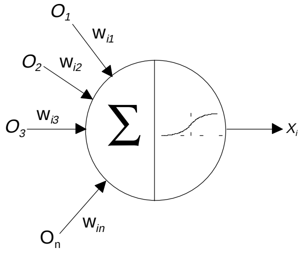

Fig. 1 : Model of an artificial neuron

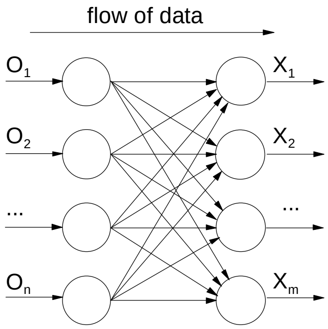

Fig. 2 : Part of a feed-forward network

If all neurons of one layer are regarded (Fig. 2), the function of the complete layer can be described as a matrix/vector multiplication

$$\underline{x} = f\left(\underline{W} * \underline{o}\right), \tag{2}$$

where o is the input vector and W the weight matrix which keeps all connection weights between two related layers. The sigmoidal function

$$f(x) = \frac{1}{1+e^{-\alpha x}} \tag{3}$$

is mostly used as the activation function in feedforward networks.

Feedforward networks are very powerful when using one or more hidden layers. The structure or topology of the network determines the class of geometry for pattern recognition, function approximation or transformation to be described by the neural network (see Fig. 3). With one layer linear separable problems can be solved. Every output neuron divides the input space into two regions. If the sigmoidal function is assumed to be very sharp (with α being very large) the function of an output neuron may be visualized by a separation line (in a 2-dimensional input space) or a hyperplane (in an n-dimensional input space). Non-linear problems cannot be solved by this type of network. In Fig.3 we can see that a two-layer network is able to solve all convex tasks. Even non-convex objects may be separated by one hidden layer networks if the problem is given in a pixel or boolean representation, but many neurons are required in general. Only with three layers arbitrary complex structures can be recognized with a reasonable number of neurons.

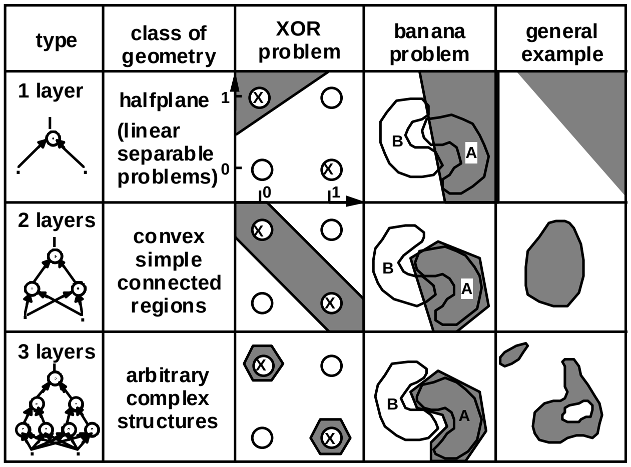

Fig.3: Capacity of multi-layer feedforward networks

To increase calculation speed of a neural network, neurons have to work in parallel. On the other hand a high flexibility concerning the structure of neural networks should be ensured. To grant both demands, only neurons within the same layer are processed in parallel, whereas the various layers are processed sequentially. The architecture of the chips and the design criteria of the system are described.

2. Selection of Neural Hardware

With different methods it is possible to reduce the computation efforts while executing neural networks. One example is an implementation of the non-linear sigmoidal activation function in a lookup table. Memory access can be done simpler and faster (less than 20 ns) than the computation of an exponential function with multiplications and additions. In the same direction it is possible to reduce the network size by eliminating useless neurons and weights near to zero. When executing the network they need not to be computed. With some applications such methods will not improve the performance. A general acceleration is only possible with parallel processing.

At the Research Center Karlsruhe neural networks are used not only for trigger purposes but also for some industrial applications. One application uses surface acoustic waves to detect and analyze unknown gases. Another project determines gas concentrations with a resistor array of 40 gas sensitive resistors. Neural networks are also used in pattern recognition tasks of pipeline crack detection. There is a variety of different applications each of them demanding for their own criteria an effective hardware solution.

Following criteria are considered to be crucial for choosing the right hardware:

1.both applicable as a PC-board and stand-alone
2.cheap
3.few peripheral devices
4.sufficient precision (at least 16 bit)

The PC-board is necessary to provide a user friendly programming tool for development. The stand-alone solution facilitates the use of the hardware acceleration in applications independent of the platform of development. The second criterion is decisive for industrial applications. This criterion applies, e.g., to the gas analyzers introduced in the former section. Point 3 is crucial for micro-systems with small space and very low power consumption. In second level trigger of particle physics experiments (e.g. calorimeters) a sufficient precision is needed because of the high dynamic range of signals.

There are especially three neuro chips available fulfilling partly the given selection criteria. The MA16 of Siemens [1], CNAPS of Adaptive Solutions [3] and ETANN of Intel [6]. MA16 mainly fails criteria 1 and 3. In the meantime SYNAPSE with four MA16 is available as PC-board [10]. To make a stand-alone solution many other chips and a micro-controller are necessary. CNAPS and ETANN fail criterion 4. CNAPS is working with 8 bit accuracy, or 16 bit with less than half the rate. Even worse, the analog ETANN computes with approximately 6 bit accuracy. Sometimes poor accuracy may be compensated by non-linear data transformations. But for on-chip training of the neural network a minimal data length of 16 bits seems to be necessary to find the global optimum.

In the following sections the alternative neuro chip is introduced to meet better the presented four criteria.

3. Design Criteria of SAND

SAND is a cascadable, systolic processor array designed for fast processing of neural networks. The neurochip SAND may be mapped on feedforward networks, radial basis function networks (RBF) and Kohonen feature maps.

Due to these most common neural network types, SAND covers about 75% of all important applications. This estimation is result of an analysis of 154 applications found in the literature. In the following the idea of the organization of the neural processor is given for the feedforward network, mostly used.

Looking at the matrix/vector multiplication introduced in the first section, it is obvious, that the weight matrix can be separated into several row vectors, see Fig.4. These vectors can be multiplied in parallel with the activation vector *o*. Each row *i* of the weight matrix corresponds to the weights connecting *n* transmitter neurons with one receiver neuron. If a layer consists of *m* neurons, a systolic array with *m* parallel working processing elements is required to get full use of a parallel architecture. One important disadvantage of this solution is the need of *m* weight memories. This fact causes high costs in peripheral components and a huge area for busses. We can think of another solution where only one weight memory is used. This single memory must be read at high speed (*m* times higher compared to the previous solution), not possible with available memories.

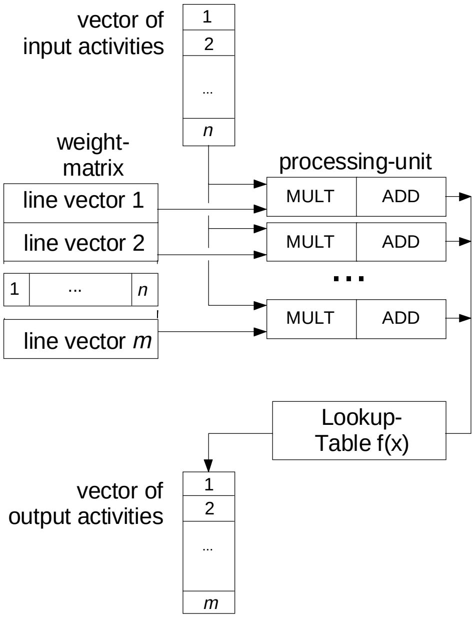

Fig. 4 : Architecture of a matrix/vector multiplier

A solution of this principal problem can be obtained by adjusting the number of input activities to the number of weights. Then a maximal usage of hardware can be ensured. This demand can be granted if several input patterns are used instead of one. The activation vector is replaced by a matrix which consists of *m* columns. In the example in Fig. 5 with *m*=4 it is shown, how incoming activities are multiplied with the corresponding weights. Values which are already transfered into SAND are marked with a grey shadow. The four processors are symbolized by a circle, a pentagon, an octagon, and a square, respectively. It can easily be seen that four processors compute 16 multiplications within 4 cycles. In every cycle only one weight and one activity have to be transfered.

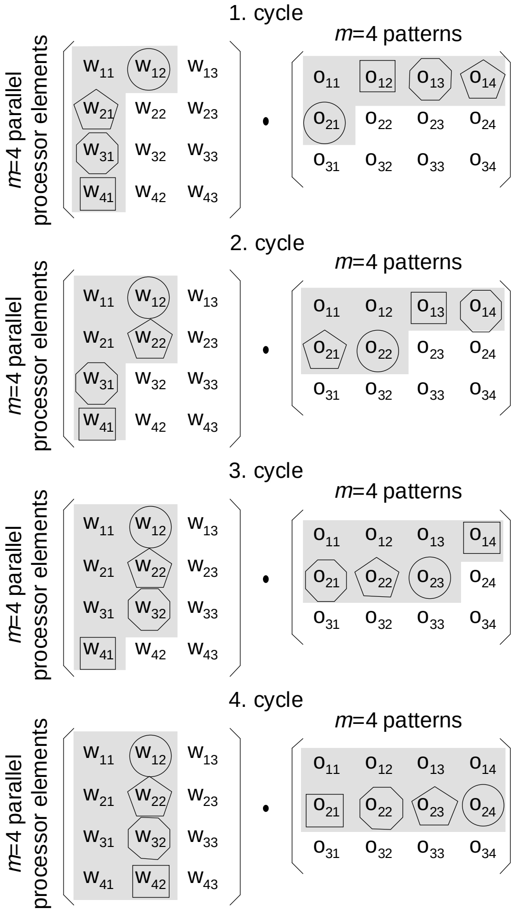

Fig. 5 : Example of processed data (m=4 patterns)

4. Architecture of SAND

Based on the design considerations of the previous chapter, the architecture of SAND was developed. Looking again at the example of Fig. 5 one can see that each of the processor elements (PE) is working four cycles with the same weight. Every fourth cycle the weight is updated so there is a continuous flow of weights on the weight bus. In the considered period of four cycles four activities are loaded into SAND's processor elements. These activities are transfered over registers from one PE to the next. There is a continuous flow of data on both the activity and on the weight bus. Due to the method data and commands are handled the architecture of SAND is a systolic processor array. In Fig. 6 the architecture of SAND is shown, which consists of four parallel processing elements each equiped with an ALU and an auto-cut module. The ALU in Fig. 6 is used for the multiplication of vectors within the matrix/matrix multiplication. Due to the accumulation of activities the width of words grow from 16 bit up to 40 bit, if the number of input neurons is limited to 512. To be compatible with external memories (16 bit), and with the width of activities outside the chip (16 bit), a window of 16 bit must be cut out of 40 bit. The position of this window may be influenced by a user-defined selection of an appropriate weight range. The cut is done in the auto-cut module which automatically checks if an over/underflow occurs. To minimize the error caused by the cut, an automatic adaption of the accuracy is performed in a second step.

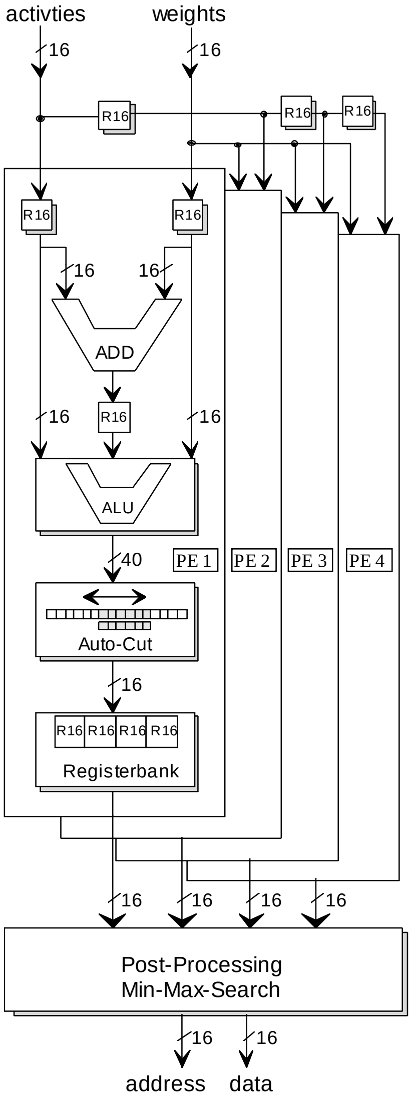

Fig. 6 : Architecture of SAND

In some cases it is important to find extremal values in the flow of output activities. Therefore, a postprocessing module is used which can work in two modes: search for a maximum or a minimum. The appropriate activation function *f(x)* is realized outside the chip with a lookup table. Some types of neural networks require both a linear function *f(x)=x* and a non-linear function like the sigmoidal function. Therefore SAND has two outputs: one for addresses of the lookup table and one for linear data.

For the calculation of expression (1) a multiplier and an adder are needed to perform a fast multiplication of vectors. To increase speed both elements are placed within a pipeline. As a first step input activities are multiplied with corresponding weights and then added to previous values. Due to the fact that four patterns are processed, four accumulation registers are required within the PEs. For some neural networks it is also necessary to calculate the Euklidian distance between two vectors. Therefore SAND's ALU is equipped with an additional adder, which is also placed in the pipeline. This feature is essentially used for Kohonen Feature Maps or Radial Basis Function Networks (RBF).

5. Structure and Operation of VME Board

With the VME neural processor board a fast (up to 800 MOPS) and universal artificial neural network (ANN) processor, simple for programming, should be provided, easily and with low expenses integrated into high energy physics (HEP) experiments with low expenses. Analysis of applications of ANN hardware in HEP experiments demonstrates that there are two large fields of such applications: the use of ANN processors for trigger systems [2,4], and for preprocessing of multi-channel measurement data (e.g. FADC output [5]).

In the first case the input data from a Data Acquisition System (DAQ) of the experiment emerge sequentially event by event. In the same order they appear at the readout bus of the DAQ. In most of the applications the data are collected from different parts of the DAQ via some specific busses and undergo preprocessing in a DAQ-specific concentrator preprocessor unit. The concentrator will be connected to the neural processor module with a simple and fast (up to 40 MHz, 2*16 bit) and non-expensive Front Panel Data Port (FPDP channel) [9]. The second field of applications represents the processing of results of multi-channel measurements. In this case the data may be delivered via VME-bus. To speed up receiving data, a DMA-transfer facility has been provided on the VME neural processor board.

A block diagram of the module is presented in Fig.7. The module consists of a 'Processing Core' built of up to four neural processor chips SAND, a Command Sequencer performing control on execution of ANN algorithms, a controller of the input data streams ‘In_data Control’ and input data buffer ‘FIFO_in’, a controller of the output data streams ‘Out_data Control’ and output data buffer ‘FIFO_out’, a VME controller with configuration memories and an FPDP and a an ECL port. The VME controller allows to operate the module on the VME-bus as slave or as master (on power-on the controller is configured automatically as VME slave with the base address set by jumpers). Four FIFO_in reorganize the input ‘event by event’ data stream into a data stream of interleaved activities of four events. Four FIFO_out’s perform the reverse transformation of the output data.

To prepare the module for automatic execution of the ANN algorithm, a VME host computer must load the ‘ANN configuration’ RAM with a description of the ANN be processed. The description represents a series of 32-bit words, describing each layer of the ANN, one word per layer. The configuration word contains the information about the number of input activities of the layer, the number of nodes in the layer, the type of operation executed on the input activities (multiply-accumulate, square-accumulate or search for min/max), the mode of transformation of the accumulated 40-bit result into 16-bit integer values, and the type of the activation function for the layer (linear or nonlinear, stored in the lookup table). Moreover, the VME-host must load the weight matrices of all used layers into the WRAM’s. The procedure of loading looks quite straight forward and simple, because the sequencer takes care of distributing the weight matrices among the WRAM’s and the user needs not to know how many SAND chips are plugged in.

During automatic execution of the ANN algorithm with incoming data, three independent processes are running simultaneously: acception of input activities for the next computation cycle, execution of the ANN algorithm in the processing core, and transfer of output activities of the previous calculations to the chosen output port (VME or ECL). Finally the type of output is selected (continuous 16 bit or with yes/no-threshold).

Table 1: Different types of VME-transfers

Transfer Mode	Transfer Rate	Transfer Type
MBLT64	73 Mb/s	Block Transfer
BLT32	40 Mb/s	
D32	20 Mb/s	Single Cycle Protocol
D16	12 Mb/s	

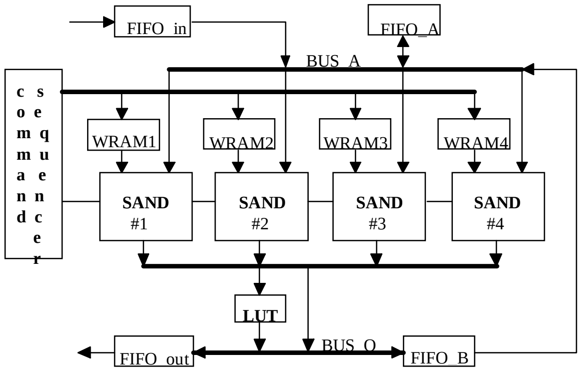

Fig. 8 : Block diagram of SAND processing engine

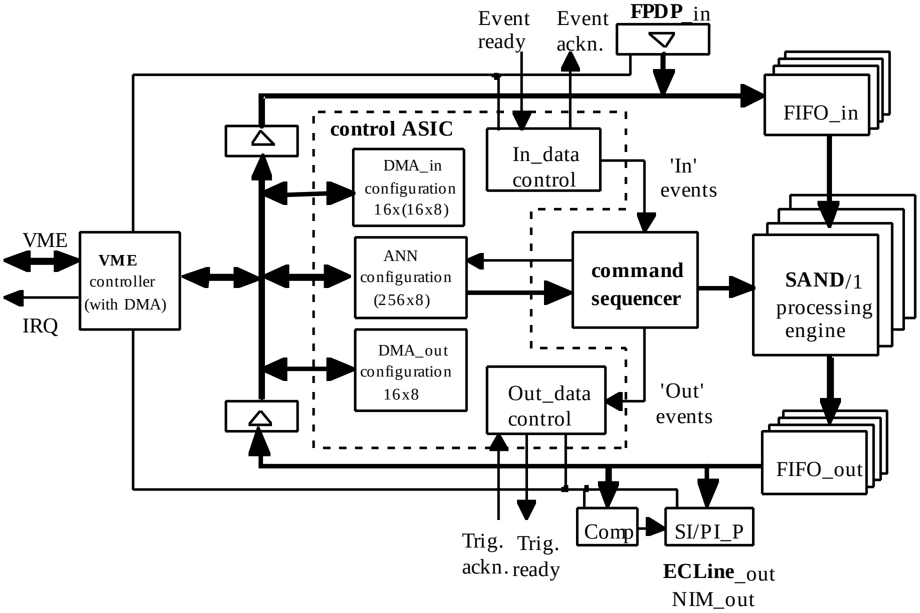

Fig. 7 : Block Scheme of a VME board for the KASCADE - experiment supporting four SAND chips.

5.1 Data stream controller

The control of the input/output data streams are configured and organized in a data stream controller implemented in one ASIC, see Fig.7. The data stream controller contains controllers, configuration and status registers for input/output data streams of SAND:

- VME-DMA input/output configuration
- FPDP input controller
- ECL/NIM output controller

The input data source may be VME-DMA or FPDP. For the output, similar options may be chosen (VME or ECL/NIM). FPDP data input allows for multi-source data collection via one multi-wire flat ribbon cable of maximum 1m length.

The controller of the input data stream can organize the autonomous collection of input activities from several DAQ slave modules situated in the same VME-crate together with the neural processor module. In this case the DMA master capability of the VME controller is used. The configuration of the DMA channels, each including the VME-starting address, local starting address, type of transfer and block length must be loaded from a VME-host during initialization of the module. To avoid a VME-timeout, the module must request the VME-bus and start readout of the DAQ slave modules only when data of the whole event are present. Since the module cannot perform a check of ready flags of the slave DAQ modules via VME-bus, the controller of the input data stream needs to be fed via a front panel LEMO- connector with an Event_ready signal (taken from a supervisor of the DAQ, for example). The data transfer rate via VME bus depends on the time response of the slave modules and on the type of VME-transfer (see Table 1).

5.2 Command Sequencer and processing core

On receiving a request from the input data stream controller, the sequencer (see Fig. 8) reads the configuration of the first layer from the ANN configuration RAM and organizes a segment by segment calculation of the layer activities. Since each SAND has four ALU's, the processor board containing four SAND chips may manage 16 neural units simultaneously. During the processing of the first 16 nodes of the first layer (first segment of the layer), the input data are taken from the FIFO_in and pushed into a circular buffer FIFO_A. Here the data are available for calculation of the next segment of the layer. Simultaneously, with moving data, the processing units compute 16 activities of the segment. After the last input activity has been taken from FIFO_in, the processing units complete the calculation of the first segment, push the results via lookup table into the buffer of the hidden layer FIFO_B and start the calculation of the next 16 activities. Input activities for the calculations are taken from FIFO_A. In a similar manner the processing core computes activities of the next layer. The total calculation time of a network consists of the time necessary to read input activities for all the segments of the network.

The architecture of SAND was adapted for processing four events in parallel. When the controller of input data stream sends less events for processing, missing events are replaced by dummy data. The results of processing these dummy data are not pushed into the FIFO_out.

6. Technical Data and Performance

SAND is manufactured in a 0.8μm CMOS process, using a sea-of-gates technology with almost 50K Gates. The packaging of SAND is a PGA with 120 signal-pins.

SAND has four parallel working processing elements (PE) on one chip, each equipped with a 16 bit fixed point multiplier and a 40 bit adder in a pipeline. Data coming from the input is passed through clocked registers from one PE to the next (Fig. 9). To make use of the parallel structure, four epoches of activities are processed in one cycle. In this way a matrix/vector multiplication is replaced by a matrix/matrix multiplication, insuring a permanent and full use of the parallel processing units and yielding 200 MOPS per chip operation speed at a cycle time of 20 ns.

The non-linear activation function is calculated by the use of a free programmable look-up table allowing for a maximum of flexibility. A controller chip, the memories, the lookup table and the SAND chip are arranged as a fixed modular unit guaranteeing the tight timing for up to 50 MHz operation (Fig. 8).

There are two well known applications of digital neural network processors in second level triggers: CNAPS[3] in H1 [2] and MA16 [1] in WA92 [4], see Tab. 2. The neural processor module based on SAND demonstrates throughput similar to the CNAPS-board and successfully competes with it when the data acquisition system is equipped with an event buffer. Moreover, the module allows processing of higher accuracy input activities. The SAND processor module shows higher throughput than the trigger module based on MA-16 due to the simultaneous processing of four events and the higher clock frequency of SAND board.

The throughput of the VME Neural processor module is limited by the time necessary to read data from the DAQ system into the module and by the calculation time (time spent by SAND chips for the calculation of output activities). The last one depends on the configuration of the network. Solid lines in Fig. 9 show the calculation time against the amount of input activities (N) for different two-layer feedforward networks (N:16:16, N:32:16 and N:64:16). Dashed lines in the figure show the readout time for different data transfer channels (DMA or FPDP) and different modes of the transfer. In the case of processing a feedforward network N:32:16, and using a VME bus for MBLT64 DMA transfer of the data, processing time is limited by the calculation time for low values of N and by the readout time for larger N (thick line in Fig.9).

Tab. 2 : Comparison of existing MA16 and CNAPS data with SAND (from simulation)

ANN structure	input activities	computation time	latency time
CNAPS	8bit/20MHz	8µs	un-
64x64x1			known
SAND	16-bit/40MHz	5.1µs	27µs
MA-16	16-bit/8MHz	5.5µs	8µs
16x5x1			
SAND	16-bit/40MHz	0.5µs	3.6µs

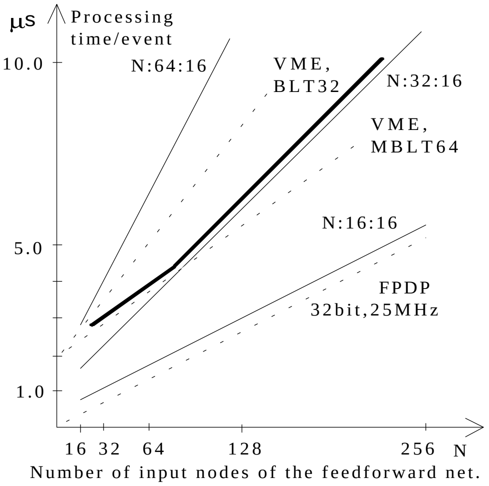

Fig. 9 : Network sizes related to various data transfers

7. Software

With the help of several software drivers it is easy to integrate the SAND VME-board into existing applications. The drivers are sets of C-functions which are adapted to operating systems like LYNX-OS and LINUX. The following table gives a brief overview of functions supported by SAND's software.

Currently a C++ class library is under development which allows users to build easily neural applications using the SAND VME-board. It is not necessary to have detailed knowledge of SAND hardware because the software integrates the board as an intelligent co-processor. Besides the possibilty of using SAND as a trigger module for high energy physics, it can also be used for acceleration of neural simulations within software packages like Stuttgarter Neural Network Simulator (SNNS). Other graphical front end tools are under development.

Tab. 3 : SAND commands

command	explanation
init_netcfg	loads and activates the configuration of a neural network
init_wram	loads the weight matrix into the weight memory
init_lut	loads the lookup table with a non-linear activation function
init_DMAcfg	loads the DMA configuration
ld_data	sends activities to SAND
rd_result	after processing has finished, data is transfered back to host
ld_CMD	load commands
rd_STAT	read status

8. Conclusion

Depending on the application several design criteria for ANN chips have to be met. These are partly different, especially in respect to the size and the type of the neural network. SAND performs feedforward networks, Kohonen feature maps and radial basis functions with comparable speed. The central processing unit of the chip was designed in a way that only few additional devices are required compared to previous designs. To facilitate a stand-alone operation of SAND, the neuron activities are buffered. Because of the modular structure, performance improvements may be achieved by adding more processing elements. Furthermore, a VME (see Fig.11) and PCI board supporting four chips of SAND (adding up to 800 MOPS) are under development.

It is the main goal of this paper to stimulate the discussion for a new generation of digital neural chips. Future developments of general purpose micro-processors like pentium P55C from INTEL, K6 from AMD, M2 from Cyrix and others have to be regarded carefully. Their MMX-instructions and the use of parallel integer units on chip enable these devices to very fast matrix multiplications. Up to now the independent parallel transfer of data is a problem so that they cannot compete with the performance of SAND. Because of many restrictions the internal pipeline organization is not appropriate for the fast computation of neural networks. Other processors (from MIPS with MDMX-instructions) or the digital signal processor TMS320C80 from TI aim to the same direction dealing with similar problems. But in near future this may change.

At FZK for medical and industrial applications, and at particle physics experiments for trigger purposes and on- and off-line data evaluation, computing power for neural network operations in the range of 1000 MOPS and more is demanded. A first silicon implementation for SAND, a semi-custom chip with 200 MOPS, is expected by the end of 1996. The chip is produced by IMS[1], Stuttgart, the PCI-board with four SAND chips is coming from INCO[2], Leipzig and for the VME-board, also with up to four SAND chips, STRUCK[3] is responsible. Faster versions using full-custom design and supporting a fast hardware learning features are under development.

9. References

[1] U. Ramacher et al., ”Design of a First Generation Neurocomputer“, in VLSI-Design of Neural Networks, eds. U. Ramacher and U. Rückert, Kluwer Academic Publishers, 1991

[2] D. Goldnez et al., Proceedings of 4_{th} International Work-shop on software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics, April 3 - 8, 1995, Pisa, Italy, pp.333-340

[3] Adaptive Solutions, CNAPS product Information, 1995

[4] C. Baldanza et al., NIM A 376 (1996) 411 and NIM A 373(1996) 261

[5] P.Cennini et al., NIM A 356 (1995) 507

[6] C.S. Lindsey and B. Denby : „A study of the Intel ETANN VLSI Neural Network for an Electron Isolation Trigger“, 1992, internal CDF Note - CDF/DOC/CDF-/Public/1850

[7] C. Kiesling et al.: „A level-2 Neural Network Trigger for the H1 Experiment at HERA“, KHEP 1994

[8] T.Fischer, H.Gemmeke, W.Eppler, A.Menchikov, S.Neusser: „Novel digital hardware for trigger applications in particle physics“ in Proc. of 2nd Workshop on Electronics for LHC Experiments, Balatonfüred, Hungary 1996

[9] Front Panel Data Port Specification (VITA 17 199x)

[10] Siemens Corp., „Neurocomputers and Tools“, SYNAPSE 2-Homepage at http://www.snat.de“

[1] IMS, Allmandring 30a, D-70569 Stuttgart, Germany

[2] INCO GmbH, Stöhrer Str. 17, D-04347 Leipzig, Germany

[3] STRUCK, Beckerbarg 6, D-22889 Tangstedt, Germany