NeuroMem Technology Reference Guide

Version 4.0.1
Revised 05/06/2016

1 Table of Contents

2 Introduction

The NeuroMem technology is a crucial enabler for cognitive computing and artificial intelligence. It is an architecture of cognitive memories which react to input patterns and can be compared to the brain because of its low power requirements, scalability and instantaneous internal communications.

A NeuroMem chip is a fully parallel silicon neural network: it is a chain of identical elements (i.e. neurons) which can store and process information simultaneously. They are addressed in parallel and have their own "genetic" material to learn and recall patterns without running a single line of code and without reporting to any supervising unit. In addition, the neurons fully collaborate with each other though a bi-directional and parallel neuron bus which is the key to accuracy, flexibility and speed performance. Indeed each neuron incorporates information from all the other neurons into its own learning logic and into its response logic. This mechanism prevents the learning of any redundant information, but also the immediate detection of novelty or potential conflicts. Another resulting achievement of the parallel architecture of NeuroMem is its constant learning and recognition time regardless of the number of connected neurons, as well as the ability to expand the size of the neural network by cascading chips.

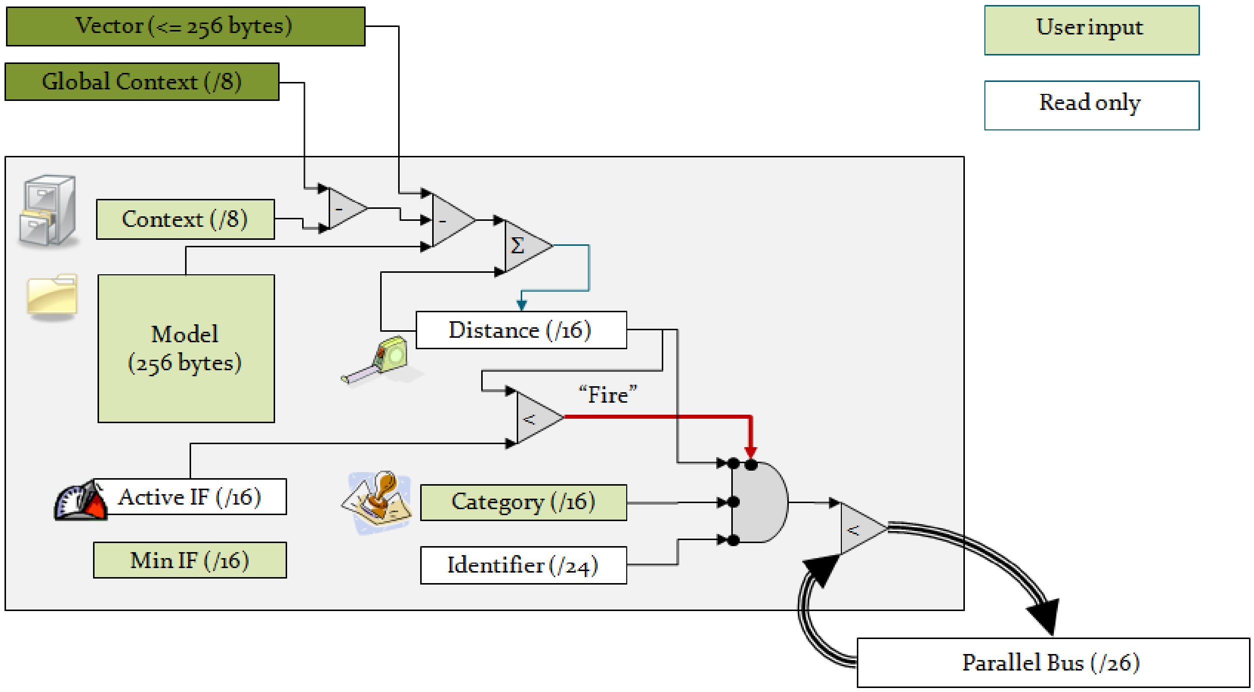
2.1 NeuroMem Key Features

- Parallel broadcast mode
  - All the neurons update their distance value simultaneously as the components of an input vector are broadcasted on their parallel bus. Upon receipt of the last component of the input vector, all neurons have calculated its distance to the reference pattern they hold in memory. If an input vector is broadcasted to a chain of 10, 100 or 1000 NeuroMem chips, their distance values are calculated and ready to be read as soon as the last component of a vector his broadcasted.
- Autonomous model generator
  - The model generator built-in the NeuroMem chip makes it possible to learn examples in real-time when they drift from the knowledge residing in the current neurons. The "novelty" examples can be stored in neurons assigned to a different context to allow a supervised verification and learning at a later time.
  - The knowledge built by the neurons is clonable since the content of the neurons can be saved and restored.
- Reactive recognition with Winner-Takes-All
  - The neurons are capable of ranking similarities between input vectors and the reference patterns they hold in memory, but also reporting conflicting responses or cases of uncertainty, reporting unknown responses or cases of anomaly or novelty.
  - The neurons order their response autonomously per increasing distance value as the host processor sends K successive read commands of the Distance register. Again, this unique feature pertains to the parallel architecture of the neurons and a patented Search and Sort process which allows them to know if other neurons have a smaller distance value without the need for a supervisor or controller.
- Fixed latency
  - The time necessary to obtain a response is independent from the number of committed neurons in the network and from their type of response. At each read command, only the neuron with the smallest distance outputs its value to the parallel bus after 19 clock cycles. If an application requires the use of a KNN with K equal to 50 for example, the distance values of the 50^{th} closest neurons are read in 50 * 19 clock cycles.
- Multiple contexts or network dynamic segmentation

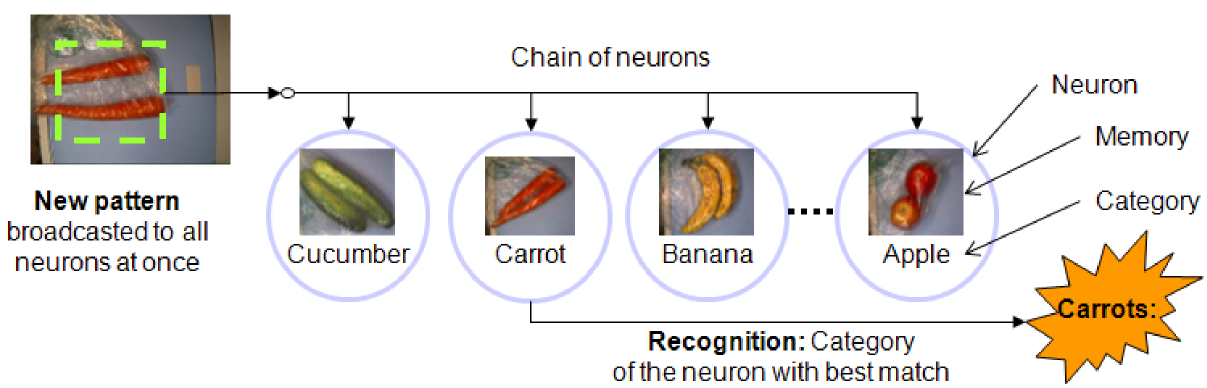
    - The ability to assign the neurons to different contexts or sub-network allows building hierarchical or parallel decision trees between sub-networks. This leads to advanced machine learning with uncertainty management and hypothesis generation.
- Multiple type of classifier
    - The neurons can behave as a KNN or RCE (class of RBF)
    - A Restricted Coulomb Energy (RCE) classifier uses Radial Basis Function as activation function. It is capable of representing complex nonlinear mappings and widely used for function approximation, time series prediction, and control.
    - A K-Nearest Neighbor algorithm (*K*NN) is a method for classifying objects based on closest training examples in the feature space. The parallel architecture of the NeuroMem chip makes it the fastest candidate to retrieve the K closest neighbors of a vector among ANY number.

3 What is a neuron?

A neuron is a cognitive and reactive memory which can autonomously evaluate the distance between an incoming pattern and a reference pattern stored in its memory. If this distance falls within a range called the influence field, the neuron returns a positive classification which consists of the distance value and the category of its reference pattern.



Although a neuron has its own processing unit to react to a pattern, it is the collective response of all the neurons which produces interesting diagnostics. When attempting to recognize a pattern, each neuron has the capability to spy the response of its counter-parts and to withdraw itself from the race if another neuron reports a smaller distance value.



3.1 Neuron part 1: A memory holding a pattern

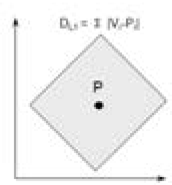
The neuron has a memory with a given capacity (256 bytes in the NeuroMem CM1K chip for example). The access to the memory cells is controlled by the neural network itself. All the neurons point to the same memory cell index at any time. The cell index is automatically incremented each time a new component is broadcasted to the neurons. It is reset to 0 when the last component is entered. Other subsidiary operations can change or reset the index as described under the chapter “Functional diagrams”.

3.2 Neuron part 2: A distance evaluation unit

The distance evaluation unit computes the distance between the incoming vector and the pattern stored in the neuron memory. This occurs in parallel for all the committed neurons each time a vector component is broadcasted to the neuron bus. The Distance value is automatically reset to 0 by writing the first component of a vector or by writing to the Network Status Register (to change the operation mode or the type of classifier).

The distance can be calculated using two norms: L1 (default) or Manhattan distance, and Lsup. The selection of a Norm depends on the application and in particular the type of patterns to classify, their possible variations between categories and the final intent of the recognition (identification, classification, anomaly detection).

Norm L1 (Manhattan distance)



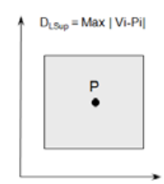
The L1 distance emphasizes the drift of the sum of the all components between V and P.

Use model

Let’s take the example of a data mining application where the profile of customers is categorized based on attributes such as age, sex, weight, skin color, date of graduation, income bracket, etc .

These attributes are expressed in different units, and some of them are actually codes rather than measurements. Still they can be assembled in a pattern vector to help classify people. In this case, the distance between an input vector and a stored prototype is not representative of any unit, but the L1 Distance gives an idea of the overall variations between them. On the other hand, the Lsup Distance is meaningless since depending on the index of the component with the highest difference, the unit can be years, dollars, codes, etc.

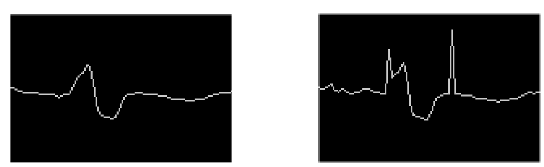
Norm L Sup



The Lsup distance emphasizes the largest drift of the same component between V and P.

Use model

Neurons can be used as a noise filter if they hold prototypes of non-noisy patterns and their Norm is set to Lsup.



In the above example, V_{noisy} shown to the right is a noisy version of vector V shown to the left. The Lsup distance between V and V_{noisy} is 50 when the L1 distance is 4900. Indeed the L1 distance increases dramatically when noisy peaks are superimposed onto the signal. Neurons trained with an L1 distance will not easily associate V_{noisy} to V. Neurons trained with the Lsup distance will make this association more easily.

3.3 Neuron Part 3: An associative recognition logic

Firing stage

The associative logic of the neuron is activated when the last component of the input pattern is received. The calculation of the distance between the input pattern and the pattern stored in the neuron is complete. If its value is less than a threshold called the neuron Influence Field, the neuron enters a status called “firing status” and outputs its category to the neuron bus.

Identified or Uncertain recognition

This output is multiplexed with the output of all the firing neurons and the result establishes immediately if the neuron is in agreement with the other firing neurons or not. If yes, the classification is labeled as Identified. If no, the classification is labeled as uncertain.

Unique Search and Sort logic

When a neuron has fired, it responds to subsequent requests for a distance and category value and when its own response becomes the smallest value. A patented “search and sort” mechanism retrieves the smallest value transmitted by all neurons at the same time on the neuron data bus. Each neuron can then use this feedback to determine if it can stay in the race of the best match in term of distance value or category value.

3.4 Neuron Part 4: A learning logic

The learning logic is activated after the associative logic when a category value is assigned to the last input pattern.

Commitment of a new neuron

Once a new pattern has been broadcasted to the network, only the “firing” neurons or the “ready-to-learn” neuron will react to a learning operation.

If no neuron fires, a new neuron gets automatically committed to hold the input pattern and its associated category. Its influence field is set to the current value of the Maximum Influence Field.

If neurons fire, a new neuron gets committed only if none of the firing neurons identifies the input pattern as belonging to the category to learn. Its influence field is set to the distance of the closest firing neuron.

Reduction of the influence field of firing neurons

If the category of a firing neuron is different from the category to learn, it will automatically reduce its Active Influence Field (AIF) to the distance value between its stored pattern and the input pattern. This distance was calculated by the neuron during the broadcast of the input pattern. The reduction of the AIF is a corrective action which will prevent the neuron from firing if the same input pattern is broadcasted to the network again.

Learning the “0” or null category

Learning a category equal to the value 0 is a special teaching instruction which cannot commit any new neuron but can force neurons which are firing erroneously to reduce their influence fields and not repeat this misfiring again the next time a same pattern is broadcasted to the neurons. Learning a category 0 is equivalent to teaching a counter example.

Degeneration of a firing neuron

If the Active Influence Field of a neuron must be reduced to a value lesser than the Minimum Influence Field value (MINIF), its AIF is set to the MINIF and the neuron is also flagged as degenerated.

A degenerated neuron behaves as any other committed neuron. If it fires, its distance and category can be read , but bit 15 of the category will be flagged to notify that the neuron was prevented from shrinking during the training phase. This can be an indication that its response should be weighted differently than the response of another firing neuron which is not degenerated.

One interest of the degenerated flag is to obtain statistics on neurons’ content at the end of a learning phase:

(1) Degenerated neurons might indicate that the learned patterns contain insufficient information to discriminate their categories. It might also pinpoint errors in the categories assigned to the input patterns.

(2) The significant number of degenerated neurons for a specific category can help establish that an additional feature should be extracted and taught under a different context to classify this category.

<u>Example 1:</u>
In an OCR application, the U and V letters can have very similar signature. It would not be a surprise that the learning of examples of V and U degenerate some neurons because an area of uncertainty between the models of U and V exists (depending on the font of the characters).

<u>Example 2:</u>
A common cause for degenerating neurons is to send contradictory learning instructions. For example, if a knowledge already holds reference patterns of the character V and an operator teaches a new V pattern as an A character by mistake, this erroneous instruction will probably degenerate some of the neurons holding a correct V pattern.

3.5 Neuron Part 5: An element in an infinite chain

A neuron is an element in a chain of neurons. The neurons are daisy-chained to compose a network of neurons, but they all receive and decode commands in parallel as described in the next chapter.
Exceptionally, neurons can be accessed individually when their network is switch from the interactive Learn and Recognize mode to the passive Save and Restore mode.

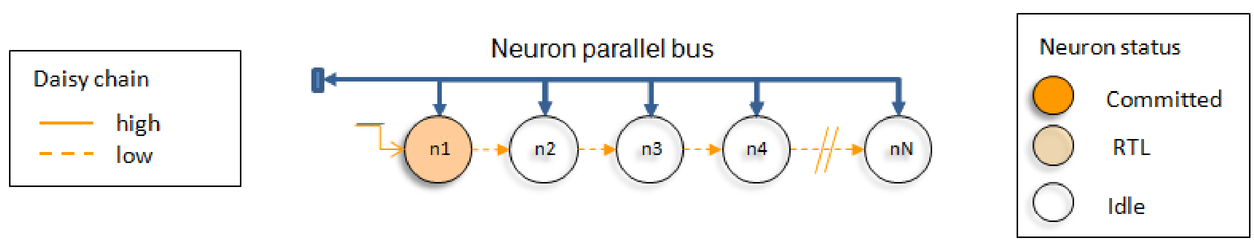
4 Network architecture

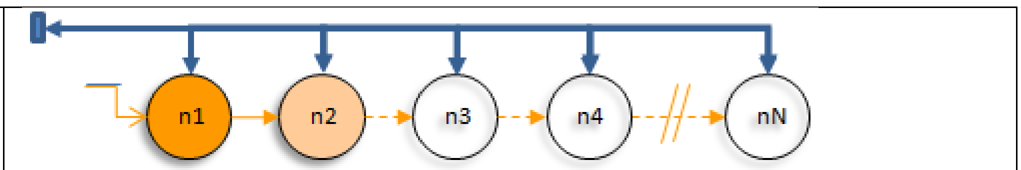
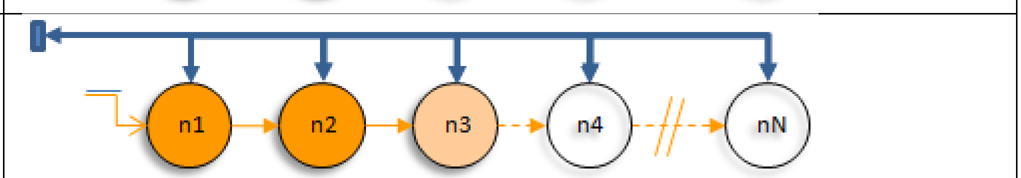
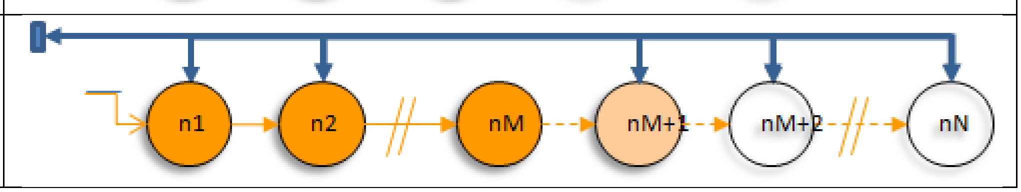
The fully parallel architecture of the NeuroMem chip is made possible because all the neurons are identical and do not require any controller or supervisor to interact with one another. They receive the same instructions and data over the neuron parallel bus and execute them at the same time. The execution of certain instructions requires that the Ready-To-Learn (RTL) and Committed neurons consult the response of one another over the neuron parallel bus. This interaction is necessary to build a consistent and adaptive knowledge.

4.1 A chain of identical neurons

At initialization, the neurons are empty meaning that they do not have any knowledge. Their status is Idle except for the first one which is Ready-To-Learn (RTL). As examples are learned by the network, neurons are progressively used to store reference patterns along with their associated category and become Committed.

The state of a neuron in the chain can be Idle, RTL or Committed. It is defined by the status of its daisy-chain-in (DCI) and daisy-chain-out (DCO) lines. The DCO of a neuron rises if its DCI is high and its category register is different from 0. As a result, the commitment of neurons is propagated automatically as examples are taught and retained. The RTL neuron also moves along until no more Idle neuron is available in the chain.



Commitment of 1st neuron	
Commitment of 2nd neuron	
Commitment of Mth neuron	

The neural network is composed of N neurons:
- M committed neurons holding a reference pattern and a category value
- 1 ready-to-learn (RTL) neuron
- N-(M-1) idle neurons

The behavior of a neuron is function of its state in the chain:

Neuron State	**Idle**	**Ready-to-Learn**	**Committed**
Behavior	Does not respond to input patterns, but updates its global registers such as Context, Minimum Influence Field and Maximum Influence Field.	Holds the last input pattern. If a category is taught and not recognized by any of the committed neurons, the RTL neuron stores the category and becomes committed. The next Idle neuron in the chain becomes the RTL neuron.	All committed neurons attempt to recognize an incoming vector. Once committed, a neuron can only shrink its Influence Field and set its Degenerated flag. Its status can return to Idle when it is instructed to forget.
Memory		x	x
Distance calculator		x	x
Associative logic			x
Learning logic		x	x
Save and Restore		x	x

4.2 Network segmentation into context

The neurons can be associated to different contexts and their use can be enabled or disabled by selecting a context value. For example, if an application uses two sensors such as a microphone to record a voice signal and a camera to record a video signal, two contexts will be used to toggle between two sub-networks: the neurons trained to recognize input vectors deriving from a voice signal and the ones trained to recognize input vectors deriving from a video signal. Furthermore, if the video camera is an outdoor camera, it might be useful to train it differently as a function of time of the day. Indeed the images will have nothing in common between day and night and the sensor might require different filter, gain and shutter adjustments. A context Day and context Night will allow training two sub networks based on the time of the day. In summary, usage of the context allows segmenting the network per family of input data. This segmentation can be based on the model of the input sensor, the settings of the input sensor, the feature extracted from the sensor data, the data length, the time of collection of the data and more.

A context is selected by writing a context value to the Global Context Register (GCR) of the chip. Any committed neuron with its own context register different from the global context register turns idle and does not participate in any learning or recognition operation. One exception: the context value 0 enables all the neurons without regards of their context.

When a neuron gets committed, its Neuron Context Register (NCR) is set to the value of the Global Context Register (GCR). Whenever the GCR is changed, all the neurons with a different context value will not attempt to recognize any input vector, nor react to the learning of a new vector. They remain idle until the GCR is changed to a value matching their context value. A GCR equal to 0 activates all the neurons regardless of their context value.

The neurons belonging to a given context define a knowledge base. If the network has neurons belonging to N different contexts, it means that it contains N different knowledge bases. Finally if the Global Context is set to the value zero, all knowledge will be used in conjunction to recognize the input pattern.

Multiple networks for combined decision criteria

Example 1: multiple zones of inspection in a part

If the good quality of a part can be decided based on the proper size, shape and position of N different components in a part, the neural network can be trained to recognize each component separately using one context per component. The final classification of the part is then based on the classification of each component and rules which can be more or less conservative or moderate. For example, a glass bottle can be shipped if its cap is properly twisted, the cap seal is in place, no contaminant is found at the bottom, and the filling level is above the minimum. It can then be diverted to a container for "Quality Assurance A" versus "Quality Assurance B" depending on its filling level.

Example 2: multiple features per object

If a material can be characterized by its colors and texture, two sets of vectors can be extracted from each sample: one describing its color distribution, and one describing its graininess. The classification of the material then relies on two contexts, or two decision spaces.

Multiple networks for hierarchical decision

The use of multiple networks trained on different features describing a same or different portions of an object allows generating hypotheses and building robust decision schemes. For example, an application with a high cost of mistake may require that at least N sub-networks produce a same classification in order consider the overall response as a positive identification.

Example 3:

A practical approach to recognizing moving targets such as vehicles in an outdoor scene may consist of learning relevant subsets of these targets such as a silhouette, a hood, headlights, a wheel, a tire, etc. Each subset is associated to a different network trained to deal with changes of scale and orientation. When analyzing the content of a new image, each network reports a map of the locations where it recognizes a pattern. The combination of these maps produces a "Transform" image which is much simpler than the original image and can be classified by a higher level context trained to verify the spatial distribution of the different subsets in order to produce a final decision. For example, it recognizes a car if it has a silhouette of type "Front View" which contains a hood, 2 headlights and 2 front views of a tire. It also recognizes a car if it has a silhouette of type "Side View" which contains 2 side views of a tire.

Example 4:

In predictive maintenance, the classification of anomalies can be processed hierarchically starting with the detection of any novelty by a top "conservative" engine (neurons of context#1) to make sure that nothing is discarded. The samples recognized as novelty trigger the use of a second engine trained to classify the data with a greater level of details. Based on its classification results, this engine can trigger other engines and so on until the novelty becomes a classified anomaly.

4.3 Save and Restore of the neurons

Once a decision space has been modeled and validated by recognizing many samples, the contents of the neurons can represent a valuable knowledge base and intellectual property.

A Save and Restore mode controlled through the Network Status Register allows to access the neurons sequentially and read/write their internal neuron registers and memory. The content of each neuron is saved in VLEN+8 bytes as follows: VLEN bytes of memory, 2 byte of context, 2 bytes of minimum influence field, 2 bytes of influence field and 2 bytes of category. The knowledge which is defined as the contents of all the committed neurons occupies 261-bytes times the number of committed neurons. Note that it is important to associate the knowledge to the description of the feature extraction techniques producing the patterns stored in the neurons and also to the network settings such as the Minimum and Maximum Influence fields used during the training. This information can be stored in a header of the knowledge file.

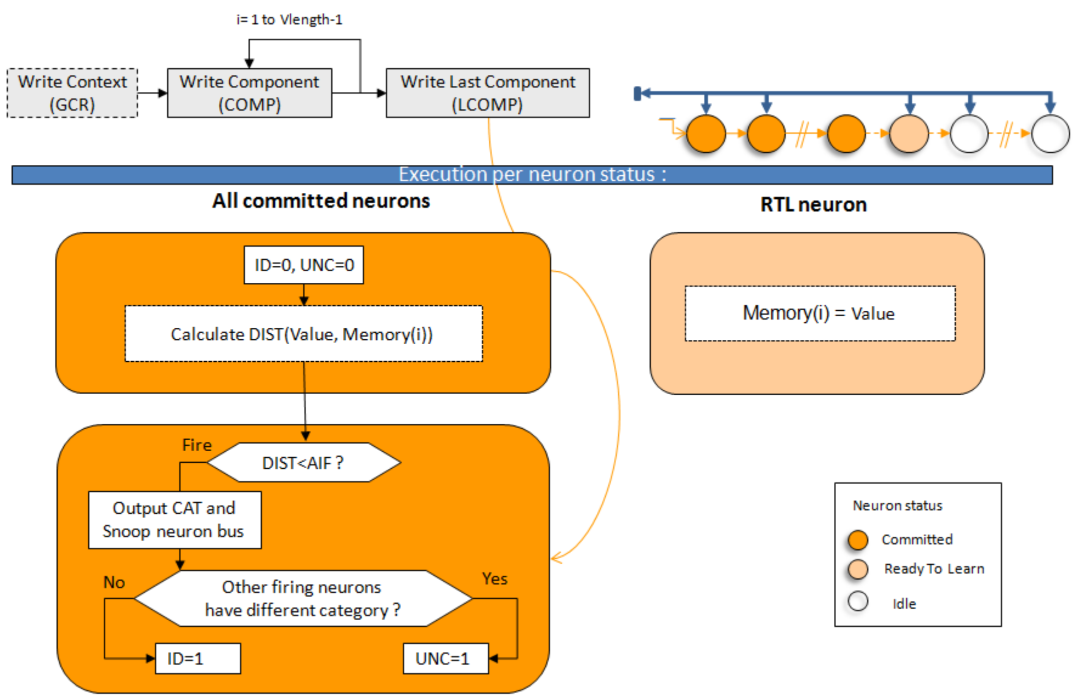
A knowledge file can be built in advance using a system interfaced to a NeuroMem chip or a simulation of NeuroMem. The VLEN+8 bytes of information necessary per neuron can be loaded at a later time. This transfer should be preceded by a clear of the neurons whenever possible. Otherwise, it becomes a merger between two knowledge bases (the one residing in the neurons and the one to load) and it must be considered cautiously since their neurons could contradict each another. If the two knowledge bases use different contexts then merging them is always safe.

5 Functional diagrams

5.1 Vector broadcasting

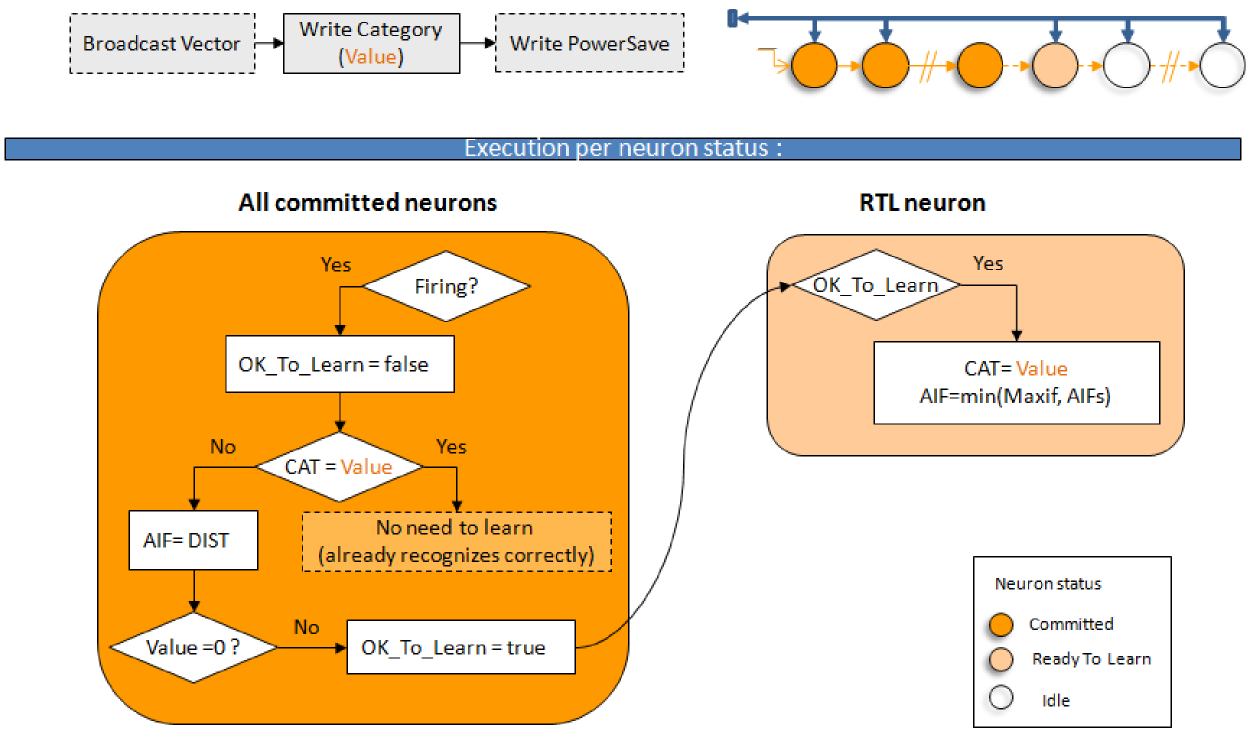
The broadcast of a vector to the neurons is made with the following sequence of operations:

1) Write Context (optional)
   a) If the new vector must be associated to a context different than the current value of the Global Context or if the distance norm must be changed
2) Loop to write the N-1 components of the input vector
   a) Write all the components of the input vector but the last one in the Ready-To-Learn. For all the committed neurons with a context equal to the Global Context, their distance register is updated after each Write Component according to the Norm in use.
3) 1Write Last Component
   a) For all the committed neurons with a context value equal to the Global Context register, their distance register is updated and represents the distance between the input vector and the prototype stored in their memory. If the distance of a neuron is less than its influence field, the neuron "fires" meaning that it is ready to respond to further inquiries such as a Read DIST or Read CAT commands. Also at the end of the Write Last Component, the entire neural network has been able to evaluate if the vector is recognized or not, and with uncertainty or not. Recognition exists is at least one neuron fires. Uncertainty exists if at least two of the firing neurons have a different category register.



5.2 Vector Learning

All the neurons have their internal learning logic and teaching a vector is as simple as broadcasting its components and then writing its category value. If the pair (vector and category) represents novelty to the existing neurons, the Ready-To-Learn neuron becomes committed. It stores the instructed category in its category register. Its influence field of the new neuron is set to the smallest distance register of the firing neurons or the Minimum Influence Field whichever is greater. If they are no firing neurons at all, its influence field is set to the current value of the Maximum influence field. The next neuron in the chain turns from idle to RTL (ready-to-learn). If there are neurons which recognized the vector with a category other than the instructed category, they automatically reduce their influence field to prevent such erroneous recognition in the future.



Global settings prior to learning

The following global registers affect the way the neurons will learn new vectors and model a decision space. Changing them should be done prior to broadcasting the vectors to learn.

Global Context	(to segment the network)
Maximum Influence Field	(to adjust conservatism)
Minimum Influence Field	(to control uncertainty domain)

Remark: the minimum and maximum influence fields must be expressed in a dimension relevant to the dimension of the input vector.

Building a knowledge independent of the training sequence

The decision space is modeled as examples are taught and consequently its shape depends on the sequence of the training examples. Indeed, their order determines the commitment of new neurons and the shrinking of existing committed neurons. This temporal dependency is not ideal and it is recommended whenever possible to learn the examples repeatedly until the decision space is stable. This condition is established when no new neuron is committed between two iterations.

The ability to execute an iterative learning requires that the training examples are stored and not streamed. The repetitive broadcast of a significant number of vectors can be time consuming, but the actual learning and modeling of the decision space triggered by the Write CAT instructions will always take a constant number of clock cycles regardless of the number of committed neurons (3 or 19 cycles depending on the recognition status).

Controlling the generalization capabilities of the network

The generalization capability of the neurons is a significant strength since it means that learning a single relevant example can be sufficient to recognize many other similar cases. However, this strength can also lead to a lack of discrimination between subtle variations and become a weakness.

Three methods can be used to control the generalization capabilities of the neurons:

1. Learn many examples representing a broad range of contextual variations and, if possible, learn them in an iterative manner as described in the previous paragraph
2. Use the category 0 to learn counter examples for the purpose of shrinking the influence field of neurons firing erroneously. This method is a dynamic and interactive correction which allows to only restrict the overgeneralization of certain neurons, as opposed to the next method based on the MAXIF which is a limitation imposed on all the neurons.
3. Reduce the value of the Maximum Influence Field (MAXIF) which is a global register setting the default influence field value of all the non committed neurons of the network. If this value is reduced, new examples have more chance to be not recognized and commit a new neuron. The smaller the MAXIF, the more committed neurons but with a smaller influence field. Refer to the Mapping Decision Space Reference guide for more information. Note that the value of the MAXIF should be set in relation to the dimension represented by the feature vectors.

Tracking the association between training vectors and neurons

Maintaining a table associating the training vectors to the committed neurons can be of interest for the analytics of the training set. This can be done very simply by reading the neuron count after each Learn operation and observing if it has incremented by 1. In such case, you know immediately that the last broadcasted vector K is stored in the memory of the neuron with the identifier equal to the neuron count (NID=NCOUNT).

Note that if the learning is iterative, the “association” of the pairs (vector K, neuron NID) can point the training vectors which were recognized at one point during the learning phase and got excluded later in the process due to the reduction of some neurons’ influence fields. For example if the table is composed of (1,1), (9,2), (4,3), this means that vector#4 was originally recognized by neuron#1, but no longer recognized after the commitment of neuron#2 due to a shrinking of neuron#1. Therefore learning vector#4 again committed neuron#3.

Accessing the last learned prototype

If you are dealing with input vectors coming from streaming or sensor data, you can retrieve the learned prototype immediately after its storage into the neuron ready-to-learn. Taking advantage of the fact the last committed neuron is in focus, its memory content can be read by switching the network to Save-and-Restore mode and reading the component of the neuron in focus. Do not forget to set back the normal mode. Otherwise retrieving the content of all the neurons can be done at a later time (see next paragraph).

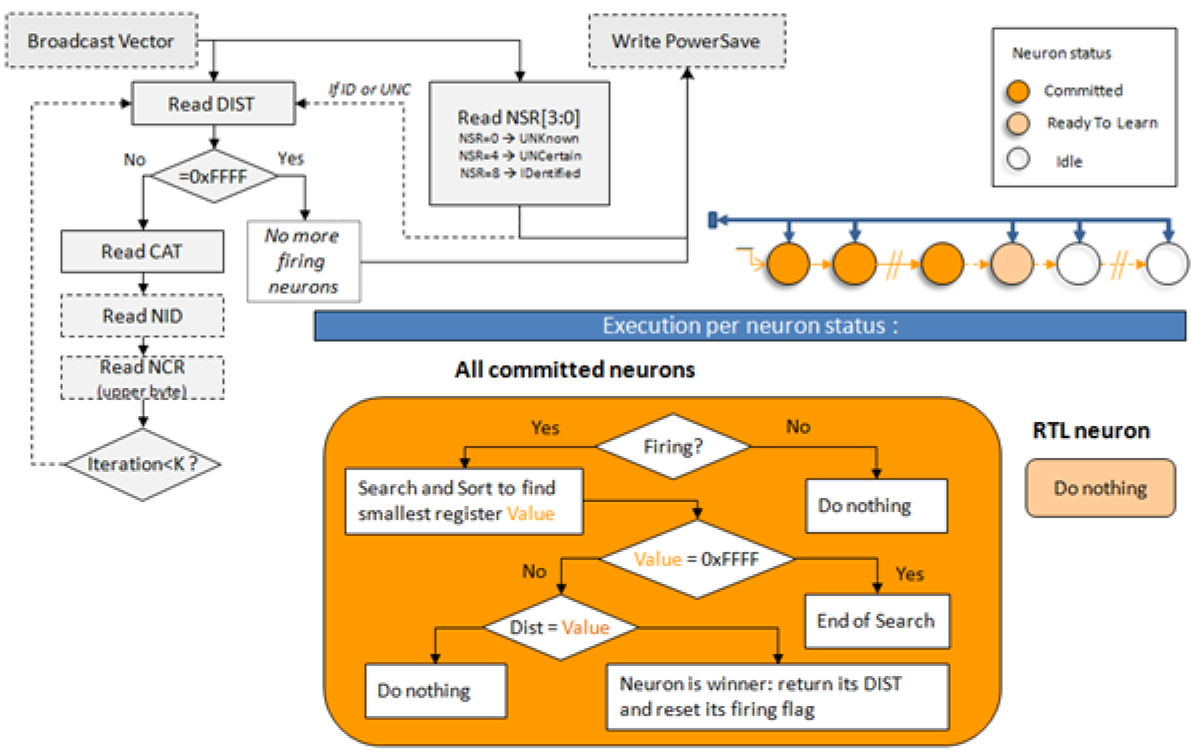
5.3 Vector Recognition in RBF mode

The recognition of a vector is readily available after its broadcast to the neurons. Indeed the neurons which identify the vector as falling within the similarity domain of the vector stored in their memory have their “fire” flag raised.

The response of the firing neurons can be accessed by a succession of (Read DIST, followed by Read CAT and optionally Read NID registers). The first distance quantifies the difference between the input vector and the neuron with the closest pattern. The category of this neuron is the category with the highest confidence level. The second distance quantifies the difference between the input vector and the neuron with the second closest pattern. The category of this neuron is the category with the second highest confidence level, and so on. In the case of the RBF classifier, all the firing neurons have been read when Read DIST returns the value 0xFFFF.

The following diagram illustrates the three levels of response which can be delivered by the neurons through the readout of the registers NSR, DIST, CAT and NID:

- Conformity, or status of the recognition (identified, uncertain or unknown)
- Best match in distance and its associated category
- All possible matches listed per increasing distance values.



Response Type 1: Conformity detection

As soon as a vector is broadcasted to the neurons, the following recognition status is known:

- Unknown: no neuron recognizes the input vector
- Identified: one or several neurons recognize the vector and agree with its category
- Uncertain: one or several neurons recognize the vector but disagree about its category

Response type 2: Best-match

The first neuron to respond is the firing neuron with the smallest distance value is equivalent to a best match. If its distance is equal to 0, it means that the vector matches exactly the prototype stored in the neuron. If the Recognition Status is Identified, the Best match is the only possible response. On the other hand, if the Recognition Status is Uncertain, other responses can be read from the neurons as described in the next paragraph.

Response type 3: Detailed matches

Examining the distance and category of all the firing neurons can be of interest to reinforce the accuracy of a decision, especially in the cases of uncertainty. The first distance quantifies the difference between the input vector and the neuron with the closest pattern. The category of this neuron is the category with the highest confidence level. The second distance quantifies the difference between the input vector and the neuron with the second closest pattern. The category of this neuron is the category with the second highest confidence level, and so on until all firing neurons have reported their response and the distance reads 0xFFFF.

If two neurons fire with the same distance but different category, their individual response are read as follows: Read Dist, Read Cat, Read Dist, Read Cat. The second Read Dist returns the same value as the first Read Dist but is necessary to access the category register of the second neuron.

If two neurons fire with the same distance and same category, only the response of the first one is read. The first Read Dist will notify both neurons to stay in query, but both will output their category at the following Read Cat and therefore exclude themselves from the next query. A second Read Dist will return the next higher distance value if applicable.

If the category value is greater than 0x8000 or 32768 (bit 15=1) you have a warning that the neuron is "degenerated". The real category value can be obtained by masking bit 15 with 0 (AND with 0x7FFF). The degenerated flag simply indicates that the neuron was prevented from shrinking its AIF to a smaller value during training and that its response should be weighted with care, or simply differently than the response of a neuron which is not degenerated.

Reading the identifier of the neuron is optional. This feature can be useful to review the content of the neuron(s) which recognize the vector.

In the case of multiple firing neurons, a global response can then be established using probability functions, dispersion of the distances, minimum number of aggregates, etc.

Example 1

The following table describes a sequence of learning and recognition operations on a set of vectors with 10 components:

Vector	Cmd	Description	1st best match	2nd best match
Vector 1= 0,1,2,3,4,5,6,7,8,9	Learn as 1	The 1st vector is stored in a first neuron		
	Reco	Its recognition generates an "exact match"	Cat=1, Dist=0	Cat=0, Dist=-1
Vector 2= 0,1,2,**6**,4,5,6,7,8,9	Reco	The 4th components of the 2nd vector is different by a value 3. All other components are identical. Still the 1st neuron recognizes the second vector.	Cat=1, Dist=3	Cat=0, Dist=-1
Vector 3= 0,1,4,3,8,5,12,7,16,9	Learn as 2	The 3rd vector is stored in a second neuron		
	Reco	Its recognition generates an "exact match" with the second neuron.	Cat=2, Dist=0	Cat=0, Dist=-1
Vector 4= 0,1,2,3,4,5,12,7,16,9	Reco	The 1st half of the 4th vector matches V1 and the 2nd half matches V3. Both neurons knowing V1 and V3 recognize V4. Neuron 2 gives the best match because it is closer with a distance of 6.	Cat=2, Dist=6	Cat=1, Dist=14
Vector 5= 0,1,2,3,4,5,6,7,	Reco	2/3 of the components matches the one of V1 and 1/3 matches V3. Neuron 1 gives the best match with a distance of 8.	Cat=1, Dist=8	Cat=2, Dist=12

Example 2

Let's take the example of a recognition where a vector is recognized by multiple firing neurons:

Distance	5	6	9	10	11	15	39
Category	8	8	7	7	7	7	5

The best match is a reference pattern of category 8 recognized with a distance 5. However if we look at the response of all the firing neurons from a statistical stand point we can observe that the first two closest neurons report a category 7, but the next four firing neurons report a category 7 with a distance which is not that much bigger. If the cost of an inaccurate recognition is low, the response of the 1st neuron with category 8 is the simplest to retrieve (and very fast). On the contrary, if the application cannot afford a false-positive, it might be wiser to involve some statistics and assume that category 7 is the dominant category and should be the one selected for a final decision. More sophisticated rules can be deployed including the analysis of the histogram of the categories, and more. Some applications might even consider the generation of a "response" vector composed of all the "firing" categories (i.e. 8,8,7,7,7,7,3,5) and to be classified by another set of neurons taught to classify the "response" vectors. The CM1K chip can handle up to 127 subsets of neurons trained for different purposes. These subsets are called Contexts.

5.4 Vector recognition in KNN mode

In KNN mode, a vector is always recognized since the classifier discards the relationship between the distance and influence field of a neuron. As a consequence, all the neurons fire and their distance and category can be read in sequence per increasing order of distance. The first distance quantifies the difference between the input vector and the neuron with the closest pattern. The category of this neuron is the category with the highest confidence level. The second distance quantifies the difference between the input vector and the neuron with the second closest pattern. The category of this neuron is the category with the second highest confidence level, and so on for K iterations.

Remark: Using the neurons as a KNN classifier does not necessarily require to learn the vectors, but rather to load them with set of reference vectors with or without category labels. This can be done in Save and Restore mode and executes in lesser time since it does not require the use of the model generator internal to the neurons.

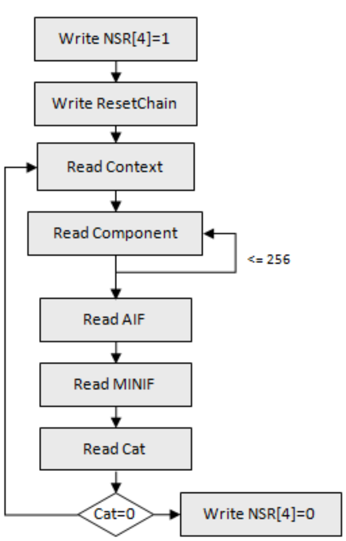
For more information on how to use the neurons as KNN classifier, please refer to the chapter "NeuroMem, high-speed KNN classifier".

5.5 Save and Restore of the neurons' content

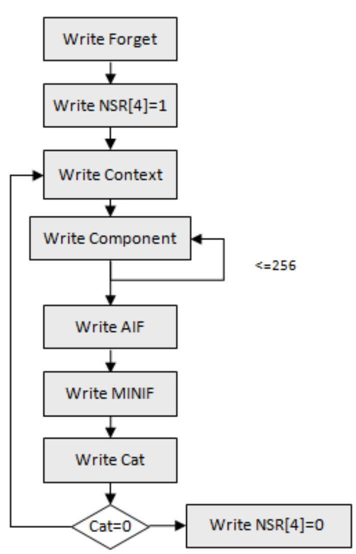
The content of the committed neurons describes a knowledge which can be saved and restored. This functionality is useful for backup purposes, but also to transfer and duplicate knowledge between NeuroMem networks.

The two functions require to set the neurons in Save_and_Restore mode and point to the first neuron of the chain. For each neuron, you can read or write its components, context, minimum influence field and active influence field in any order, except for the category register which must be read or written last to point to the next neuron in the chain. Finally, when the neurons have been saved or restored the last operation consists of setting the neurons back to their normal operation mode.

Save sequence



Restore sequence



In both sequences, once the neurons are set to the Save and Restore mode by writing bit 4 of the NSR to 1, the Category register must be the last register to be read or written per neuron since access to this register increments the neuron pointer automatically. The order in which the other neuron registers (COMP, AIF and Context) are read is not important.

Note that the LCOMP register is not used. Indeed in Save and Restore mode, the neurons are passive and simply write or return the value of the requested register. The notion of Write LCOMP which triggers the interaction between all the neurons in Learn and Recognition mode does not exist.

In the Restore sequence, the initial Write Forget which resets the category of all the committed neurons to 0 is optional. If you do not execute it, this means that you are not loading a new knowledge but rather appending knowledge to an existing one. Please note that this operation requires caution and a good understanding of the contents of the neurons. Indeed after writing bit 4 of the NSR register to 1, the neurons are set to the "passive" Save and Restore mode and do not interact with one another to learn data and adjust their influence fields. Unless the knowledge to restore describes neurons with a different context than the neurons already committed in the network, restoring without forgetting can be risky and create a corrupted or inconsistent knowledge base.

Saving and restoring the MINIF of each neuron is necessary if it is known that additional training will be done at a later time to complete or expand the knowledge. This will ensure that the degenerated status of the neurons is properly flagged if appropriate.

Warning about merging knowledge: There are few cases where loading several knowledge bases into a same chain of neurons can be relevant. One example consists of merging neurons' contents associated to different contexts and thus trained independently. If you are very cautious and clearly understand the consequences of appending the content of neurons to a knowledge already residing in a chip, you can discard the Write Forget command. In such case the neuron pointed after the Write NSR will be the first neuron available in the chain or the RTL neuron.

Reading the contents of a single specific neuron

Reading the contents of a specific neuron is made in the following order:

- The first operation consists of setting the neurons in Save_and_Restore mode and pointing to the first neuron of the chain
- In order to point to the i^{th} neuron in the chain, (i-1) consecutives Read CM_CAT are necessary
- You can then read the i^{th} neuron's components, context, minimum influence field and active influence field in any order. The category register must be read last because the instruction automatically points to the next neuron in the chain.
- Finally, the last operation consists of setting the neurons back to the normal mode.

6 The control registers

- Under Normal operations, the neurons can learn and recognize patterns as a K-Nearest Neighbor (KNN) or Radial Basis Function (RBF) and more precisely a Restricted Coulomb Energy (RCE) classifier.

Under the SR mode, the automatic model generator and search-and-sort logic are disabled. The neurons become dummy memories but can be read or written in the least amount of time. This SR mode is essential to transfer knowledge bases between hardware platforms, or make backup prior to learning additional examples.

The following table describes the 15 registers controlling the entire behavior of the neurons under the Normal and Save-and-Restore mode.

Abbreviation	Register	Normal mode	SR mode
NSR	Network Status Register Bit[1:0], reserved Bit[2], UNC (Uncertain) Bit[3], ID (Identified) Bit[4], SR mode Bit[5], KNN classifier	The ID and UNC bits are updated internally after each Write Last Comp command. ID is high if all firing neurons report the same category. The ID line is subject to an Erratum described at the end of this manual. UNC is high if several neurons fire but disagree with the category. KNN must be set to 0 while learning. Indeed, any pattern would be recognized whatever its distance from a neuron and the learning will only create a single neuron per new category.	Writing Bit 4 to 1 switches the chain of neuron to SR mode and points directly to the RTL neuron.
GCR	Global Context and also partial identifier of the RTL neuron Bit [6:0]= Context Bit[7]= Lsup Norm Bit[23:16]= Identifier[23:16]	Context in use for any new learning or recognition If the Norm is not set to LSUP, the default is the L1 Norn or Manhattan distance. Accessing the 3rd upper byte of the RTL neuron is needed if the chain of neurons is larger than 65535 neurons.	N/A

Abbreviation	Register	Normal mode	SR mode
MINIF	Minimum Influence Field	Value in use for any new neuron commitment	Value of the pointed neuron at the time it was committed
MAXIF	Maximum Influence Field	Value in use for any new neuron commitment	N/A
NCR	Neuron Context Register	Bit[15:8]=0x00 Bit[7:0]= Identifier [23:16] of the RTL neuron	Value of the pointed neuron Bit[15:8] = Identifier [23:16] Bit[7]= LSUP Norm Bit[6:0]= Context [0, 127]
COMP	Component Bit[15:8] = unused Bit[7:0]= byte component	Each Write COMP stores the component at the current INDEXCOMP value and updates the DIST register of the committed neurons with NCR=GCR and also of the RTL neuron. INDEXCOMP is automatically incremented.	After each Read or Write, moves to the next INDEXCOMP of the pointed neuron
LCOMP	Last Component Bit[15:8] = unused Bit[7:0]= byte component	Write LCOMP stores the component at the current INDEXCOMP value and updates the DIST register of the committed neurons with NCR=GCR and also of the RTL neuron. INDEXCOMP is set to 0. The ID_ and UNC_ lines are updated to report the recognition status. If ID_ line is low, the “identified category” is available on the DATA bus.	N/A
INDEXCOMP	Component index Common index pointing to the neurons’ memory	Write INDEXCOMP moves to a specific index value, but does not reset the DIST register. This value is incremented automatically after each Read COMP or Write COMP. It is reset after a Write LCOMP.	

Abbreviation	Register	Normal mode	SR mode
DIST	Distance register between [0, 65535] DIST=0 means that the vector matches exactly the model of the firing neuron. The higher the distance, the farther the vector from the model.	This register is updated by the neuron during the broadcast of components (Write COMP and Write LCOMP) Read DIST returns the distance of the top firing neuron. This “winner” neuron pulls out of the race, so the next Read Dist will ne answered by the next top firing neuron, etc. DIST=0xFFFF means that there are no more firing neurons. Must be read after Write LCOMP and before Read CAT	N/A
CAT	Category register Bit 15= Degenerated flag (read-only) Bits [14:0]= Category value between 0 and 32766 (0x7FFE) CAT greater than 32768 means that the responding neuron is degenerated. The value must be masked with 0x7FFF to report the original category of the neuron.	Write CAT of 0 does not commit a new neuron, but may force existing committed neurons to reduce their influence fields. Read CAT returns the category of the top firing neuron CAT=0xFFFF means that there are no more firing neurons Must be read after the DIST register except if the ID_ line is low and the NID register does not need to be read after the CAT register.	Category of the pointed neuron Read or Write CAT automatically moves to the next neuron index in the chain.
AIF	Active Influence Field	This register is updated automatically by all the firing neurons during learning operations (i.e. Write CAT)	Influence field of the pointed neuron

Abbreviation	Register	Normal mode	SR mode
NID	Neuron Identifier or index of the neuron in the chain. Bit[15:0]= 2 lower bytes of a 3-bytes neuron identifier. The upper byte is stored in the NCR register. Its access is only necessary when the chain of neurons is larger than 65535.	This register is assigned automatically when the RTL neuron gets committed after a Write CAT. Read NID returns the identifier of the firing neuron with the least distance and least category. It must be read after a Read CAT. (1)	This register is assigned automatically when the pointed neuron gets assigned a category different from 0 with a Write CAT.
POWERSAVE	PowerSave mode Writing this register reset the DATA lines to a tri-state mode and ensures that they do not draw current from the pull-up resistors.		
FORGET	Uncommit all neurons by clearing their category register.	Note that the neuron's memory is not cleared, but its index is reset to point at the first component. Also reset the MINIF, MAXIF and GCR to their default values.	N/A
NCOUNT	Count of committed neurons Bit[15:0]= 2 lower bytes of the count	NCOUNT=0xFFFF means that all neurons of the chain are committed. If the chain of neurons is greater than 65535 neurons this can also means that 65535 neurons are indeed committed. Reading the upper byte of the NCR register can extend the count to a 3 bytes value.	Index of the neuron pointed in the chain. Write RESETCHAIN points to the first neuron. If it is committed, NCOUNT will be equal 1, otherwise 0.
RESET CHAIN		N/A	Points to the first neuron of the chain.

6.1 Neuron behavior per status per instruction

The following table describes how the memory of the neurons is updated depending on its state in the chain of neurons.

Memory	Idle neuron	Ready to Learn neuron	Committed neuron
Component 0		Takes the value of the 1st Write COMP occurring after a Write LCOMP.	Can only be changed by a reset or restore operation. Reset the distance register
		The memory index is incremented by 1 to point to the next component.	The memory index is incremented by 1 to point to the next component.
Component 1		Takes the value of the next Write Comp or Write LCOMP.	Can only be changed by a reset or restore operation.
		The memory index is incremented by 1 after a Write Comp, or is reset to 0 after a Write LCOMP.	The memory index is incremented by 1 after a Write Comp, or is reset to 0 after a Write LCOMP.
…			
Component MAXVECLENGTH		Takes the value of the next Write Comp or Write LCOMP. The memory index is reset to 0.	Can only be changed by a reset or restore operation.

The following table describes how the registers of the neurons is updated depending on its state in the chain of neurons.

Registers	Idle neuron	Ready to Learn neuron	Committed & nselect neuron
Context	Takes the value of the Write GCR.	Takes the value of the Write GCR.	
		Current value is saved if the neuron gets committed after a Write CAT.	Can only be changed by a reset or restore operation.
Minimum Influence Field	Takes the value of the Write MINIF.	Takes the value of the Write GCR.	
Maximum Influence Field	Takes the value of Write MAXIF.	Takes the value of the Write GCR.	
Distance			The difference between the pointed Component and the input value is accumulated after each Write Comp or Write LCOMP.
Category		Value is written if no committed neuron fires and has its own category equal to value. The neuron status switches from RTL to Committed.	
Active Influence Field			Inherits the smallest distance value of the firing neurons

Commands changing the RTL neuron in chain

Memory cell index change	Normal mode	Save and Restore mode
Write COMP	Index + 1	Index + 1
Write LCOMP	Index =0	
Write INDEXCOMP	Index=k	Index=k
Write TESTCOMP		Index + 1
Write NSR	Index=0	Index=0
Write CAT		Index=0
Read CAT		Index=0