COGNIMEM
Technical Brief

Programming the CM1K made easy

This Technical Brief describes how easy it is to learn and recognize patterns with a CogniMem neural network, save and restore the knowledge built at a given time.

One of the key benefits of the CogniMem parallel architecture is that the programming of the operations accessing the neurons is 100% invariant to the size of the neural network whether it is composed of a single CM1K chip or a chain of 10, 100, 1000 or more CM1K chips. The speed performance of the learning and recognition operations will also be invariant to the network size.

15 Registers and 4 simple operations

The interface to a CogniMem neural network comes down to 4 main operations:

- Learn a vector
- Recognize a vector
- Save the knowledge built by the neurons
- Load a knowledge into the neurons

These operations are executed through Read and Write commands to the 15 neuron registers of the CM1K chip as described in the following paragraphs. For more details the CogniMem Reference Guide will give you information about the knowledge building concept and methodology, the usage of multiple neural networks, and more.

Register	Description	Addr	
COMP	Component	0x01	RW
LCOMP	Last Component	0x02	RW
DIST	Distance	0x03	R
CAT	Category	0x04	RW
AIF	Active Influence Field	0x05	R
NCR	Neuron Context Register	0x00	RW
NID	Neuron Identifier	0x06	R
NSR	Network Status Register	0x0D	RW
GCR	Global Context Register	0x00	RW
MINIF	Minimum Influence Field	0x06	RW
MAXIF	Maximum Influence Field	0x07	RW
FORGET	Clear the neurons	0x0F	W
NCOUNT	Committed neurons	0x0F	R
RESETCHAIN	Reset the neuron chain	0x0C	W

In the following examples, we will refer to pattern vectors with a length L where L can range between 1 and 256. The L components of a vector are annotated vector[i] with i ranging from 0 to L-1.

Learning a pattern

Learning a pattern vector consists of broadcasting its components to the neurons and assigning it a category. Optionally you can read the number of committed neurons to observe if a new neuron was committed.

```
For (i = 0; i<L-2, i++) Write COMP, Vector(i);
Write LCOMP, Vector(L-1)
Write CAT, Category-to-learn
Read NCOUNT, Ncount
```

Ncount will not be incremented if an existing neuron already recognizes Vector as belonging to Category-to-learn. However, the content of the committed neurons may have changed and in particular their influence fields.

Recognizing and classifying a pattern

Recognizing a pattern vector consists of broadcasting its components to the neurons and reading the response of the "firing" neurons, meaning which recognize the vector as similar to the model they hold in memory. The response of a firing neuron can be limited to its distance and/or category. It can also include its identifier number which can then be used to retrieve its content for example.

```
For (i = 0; i<L-2, i++) Write COMP, Vector(i);
Write LCOMP, Vector(L-1)
Do
{
    Read DIST, Output-distance
    Read CAT, Output-category
    Read NID, Output-identifier
} Loop Until (Output-distance == 0xFFFF)
```

Save the Knowledge Built by the Neurons

Reading the knowledge (to save it to file for example) consists of switching the neural network to Save-and-Restore mode, reading the memory and registers of all the committed neurons, and switching the network back to Learn-and-Recognize mode.

```
Write NSR, 16
Write RESETCHAIN, 0
Do
{
    For (i = 0; i<L, i++) Write COMP, Vector(i);
    Write NCR, Neuron_Context
    Write AIF, Neuron_InfluenceField
    Write CAT, Neuron_Category
} Loop Until (Neuron_Category==0)
Write NSR, 0
```

Load a Knowledge into the Neurons

Loading a pre-defined knowledge to the neurons consists of switching the network to Save-and-Restore mode, writing the memory and registers of the neurons, and switching the network back to Learn-and-Recognize mode.

```
Write FORGET, 0
Write NSR, 16
Write RESETCHAIN, 0
For (j=0; j<Number-of-Vectors; j++)
{
    For (i = 0; i<L, i++) Write COMP, Vector(i);
    Write NCR, Neuron_Context
    Write AIF, Neuron_InfluenceField
    Write CAT, Neuron_Category
}
Write NSR, 0
```

Reading the content of a specific neuron

As another example, reading the content of a specific neuron with index k in the chain consists of switching the network to Save-and-Restore mode, pointing to the neuron k, reading its memory and registers, and switching the network back to Learn-and-Recognize mode.

```
Write NSR, 16
Write RESETCHAIN, 0
For (i = 0; i<k-1, i++) Read CAT, Dummy
For (i = 0; i<L, i++) Read COMP, Vector(i);
Read NCR, Neuron_Context
Read AIF, Neuron_InfluenceField
Read CAT, Neuron_Category
Write NSR, 0
```

Timing information

Reading and writing the neuron registers takes one system clock cycle with the exception of Write LCOMP (3cc), Read CAT (3 or 19 cc), Read DIST (18cc). The following table summarizes the execution time of the 5 basic commands:

Operation	Clock cycles
Broadcast a vector of length L	L+3
Learn the last vector	18
Recognize the last vector	37 (per firing neuron)
Read Kth neuron	3+(K-1)+(L+3)
Save N neurons	3+(L+3)*N
Restore N neurons	3+(L+3)*N

Conclusion

All APIs delivered by CogniMem Technologies include these basic functions and more.