

# CM1K

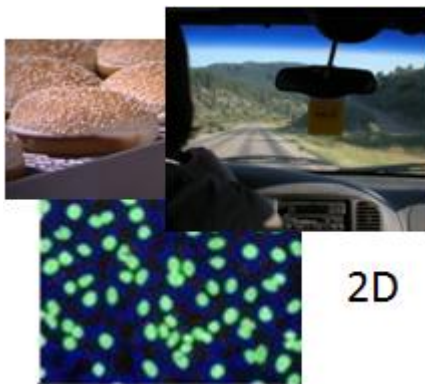
## Pattern Recognition Chip with 1024 Neurons in Parallel

### Introduction

CM1K is the first ASIC version of the CogniMem neural network. It features 1024 neurons working in parallel, capable of learning and recognizing patterns of up to 256-bytes in a few microseconds. The two non-linear classifiers (RBF and KNN) supported by the CM1K can classify patterns while coping with ill-defined data, unknown events, and changes of contexts and working conditions. The parallel architecture of the CM1K allows daisy-chaining many chips together to increase the size of the neural network in increments of 1024 neurons. Its low pin-count and low power consumption make it an ideal companion chip for smart sensors and cameras. Its cascadability makes it an ideal candidate for large data mining systems.

### Features

- Recognizes one vector among 1024, or any number, in
- 10 $\mu$ Sec with a 27 Mhz clock
- Recognition time is independent from the number of models
- Learns a vector in 10 $\mu$ Sec
- Saves and loads models (i.e. your knowledge base)
- Simple RTL instructions (fewer than a dozen registers)
- Parallel and serial communication
- Optional digital input bus directly interfaced to a recognition stage



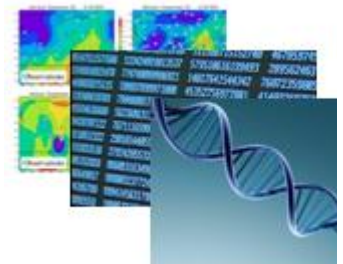
#### **Image Recognition**

- Parts inspection
- Object recognition
- Face recognition
- Target tracking and identification
- Video monitoring
- Gaze tracking
- Medical imaging, satellite imaging
- Smart motion detection, kinematics
- More...



#### **Signal Recognition**

- Speech recognition
- Voice identification
- Radar identification
- EKG, EEG monitoring
- Sonar identification
- Spectrum recognition
- Flight analysis
- Vibration monitoring
- More...

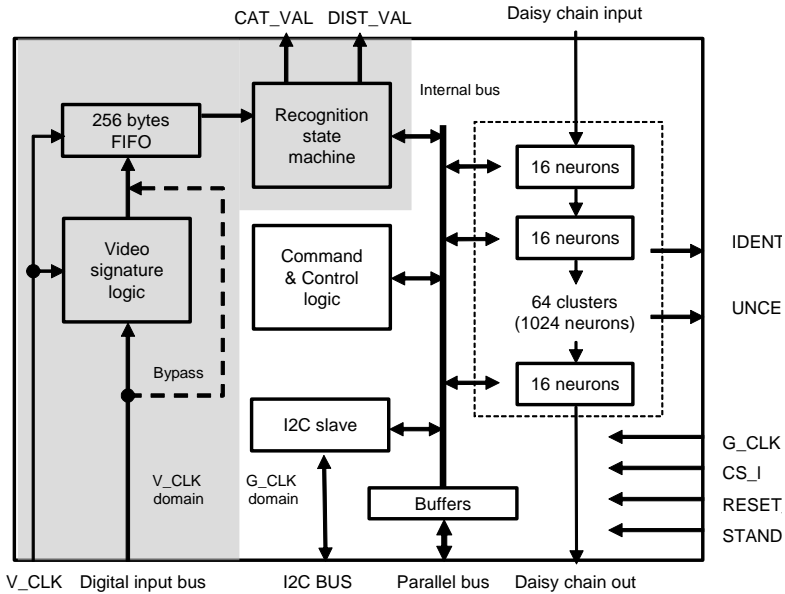


#### **Data Mining**

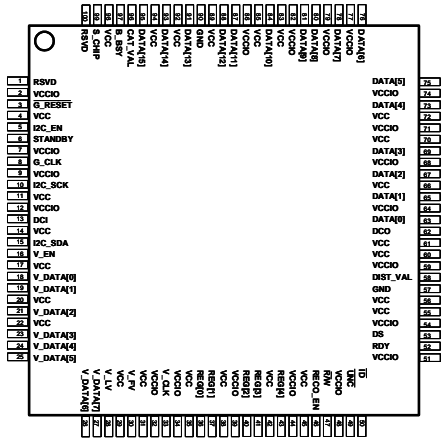
- Cryptography
- Genomics
- Bio Informatics
- Fingerprint identification
- Unstructured data mining
- CRC
- More...

# Data Sheet

CogniMem has a very simple architecture: it is a chain of identical neurons operating in parallel. A neuron is an associative memory which can autonomously compare an incoming pattern with its reference pattern. During the recognition of an input vector, all the neurons communicate briefly with one another (for 16 clock cycles) to find which one has the best match. In addition to its register-level instructions, CM1K integrates a built-in recognition engine which can receive vector data directly through a digital input bus, broadcast it to the neurons and return the best-fit category 3 microseconds later. In the case of a video input signal, CogniMem can optionally extract a 1D vector from 2D video data.



Functional diagram



**Connectivity**

- ✓ Parallel bus (DS, RW\_, 5-bit register, 15-bit data)
- ✓ Serial bus I2C (100 kbit, 400 kbit and 3.3Mbit)
- ✓ Network expansion by daisy-chaining multiple chips

**Electrical/mechanical**

- ✓ 300 mW @ 27 Mhz
- ✓ 3.3 V I/O operation 1.2 V core supply
- ✓ 100-pin TQFP package (dies available)
- ✓ 0.13 μm technology – die size 8 x 8 mm
- ✓ System clock: 27 Mhz for a single chip; <13.5 Mhz for multiple-chip configuration

**Neural Network**

- ✓ Patented parallel architecture with 1024 neurons
- ✓ RCE (Restricted Coulomb Energy) neural network
- ✓ Two classifiers:
  - Radial Basis Function (RBF)
  - K-Nearest Neighbor (KNN) classifier
- ✓ Vector data: up to 256 bytes
- ✓ Classification status: ID'd., Uncertain or Unknown
- ✓ Categories: up to 32768 values
- ✓ Distance calculation: L1 or LSup distance norms
- ✓ Sub-networks: up to 127 context values
- ✓ Trained by example
- ✓ Active learning, supervised and unsupervised
- ✓ Recognition stage for direct digital input

**Timings**

- ✓ 27 Mhz with default vectors of 256 bytes
- ✓ Learning time 10 μsec (275 cc)
- ✓ Recognition status in 8 μsec (257 cc)
- ✓ Best match in 11 μs (275 cc)
- ✓ Subsequent matches in 1.3 μsec/match (35 cc)
- ✓

**Built-in recognition stage for digital input bus**

- ✓ V\_clock up to 27 Mhz
- ✓ 8-bit data
- ✓ 1 sync. line for vector input
- ✓ 2 sync. lines for video input
- ✓ Best match 275 clock cycles after vector input
- ✓ 2 output strobe signals (Cat\_Val, Dist\_Val)