# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities




WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

## Interested in publishing with us?
## Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



7

# Algorithms for CAD Tools VLSI Design

K.A. Sumithra Devi
*R.V. College of Engineering, Bangalore*
*Vishweshvaraya Technological University, Karnataka*
*India*

## 1. Introduction

Due to advent of Very Large Scale Integration (VLSI), mainly due to rapid advances in integration technologies the electronics industry has achieved a phenomenal growth over the last two decades. Various applications of VLSI circuits in high-performance computing, telecommunications, and consumer electronics has been expanding progressively, and at a very hasty pace. Steady advances in semi-conductor technology and in the integration level of Integrated circuits (ICs) have enhanced many features, increased the performance, improved reliability of electronic equipment, and at the same time reduce the cost, power consumption and the system size. With the increase in the size and the complexity of the digital system, Computer Aided Design (CAD) tools are introduced into the hardware design process. The early paper and pencil design methods have given way to sophisticated design entry, verification and automatic hardware generation tools. The use of interactive and automatic design tools significantly increased the designer productivity with an efficient management of the design project and by automatically performing a huge amount of time extensive tasks. The designer heavily relies on software tools for every aspect of development cycle starting from circuit specification and design entry to the performance analysis, layout generation and verification. Partitioning is a method which is widely used for solving large complex problems. The partitioning methodology proved to be very useful in solving the VLSI design automation problems occurring in every stage of the IC design process. But the size and the complexity of the VLSI design has increased over time, hence some of the problems can be solved using partitioning techniques. Graphs and hyper-graphs are the natural representation of the circuits, so many problems in VLSI design can be solved effectively either by graph or hyper-graph partitioning. VLSI circuit partitioning is a vital part of the physical design stage. The essence of the circuit partitioning is to divide a circuit into number of sub-circuits with minimum interconnection between them. Which can be accomplished recursively partitioning the circuits into two parts until the desired level of complexity is reached. Partitioning is a critical area of VLSI CAD. In order to build complex digital logic circuits it is often essential to sub-divide multi –million transistor design into manageable pieces. The presence of hierarchy gives rise to natural clusters of cells. Most of the widely used algorithms tend to ignore this clustering and divide the net list in a balanced partitioning and frequently the resulting partitions are not optimal.

The demand for high-speed field-programmable gate array (FPGA) compilation tools has escalated in the deep-sub micron era. Tree partitioning problem is a special case of graph

partitioning. A general graph partitioning though fast, is inefficient while partitioning a tree structure. An algorithm for tree partitioning that can handle large trees with less memory/run time requirement will be a modification of Luke's algorithm. Dynamic program mining based tree partition, which works well for small trees, but because of its high memory and run time complexity, it cannot be used for large trees. In order to optimize above mentioned issues this chapter concentrates on different methodologies starting with Memetic Approach in comparison with genetic concept, Neuro-Memetic approach in comparison with Memetic approach, then deviated the chapter to Neuro EM model with clustering concept. After that the topic concentration is on Fuzzy ARTMAP DBSCAN technique and finally there is a section on Data mining concept using two novel Clustering algorithms achieving the optimality of the partition algorithm in minimizing the number of inter-connections between the cells, which is the required criteria of the partitioning technique in VLSI circuit design. Memetic algorithm (MA) is population based heuristic search approach for combinatorial optimization problems based on cultural evolution. They are designed to search in the space of locally optimal solutions instead of searching in the space of all candidate solutions. This is achieved by applying local search after each of the genetic operators. Crossover and mutation operators are applied to randomly chosen individuals for a predefined number of times. To maintain local optimality, the local search procedure is applied to the newly created individuals.

Neuro-memetic model makes it possible to predict the sub-circuit from circuit with minimum interconnections between them. The system consists of three parts, each dealing with data extraction, learning stage and result stage. In data extraction, a circuit is bipartite and chromosomes are represented for each sub circuit. Extracted sequences are fed to Neuro-memetic model that would recognize sub-circuits with lowest amount of interconnections between them.

Next method focuses on the use of clustering k-means (J. B. MacQueen, 1967) and Expectation-Maximization (EM) methodology (Kaban & Girolami, 2000), which divides the circuit into a number of sub-circuits with minimum interconnections between them, and partition it into 10 clusters, by using k-means and EM methodology. In recognition stage the parameters, centroid and probability are fed into generalized delta rule algorithm separately.

Further, a new model for partitioning a circuit is explored using DBSCAN and fuzzy ARTMAP neural network. The first step is concerned with feature extraction, where it uses DBSCAN algorithm. The second step is classification and is composed of a fuzzy ARTMAP neural network.

Finally, two clustering algorithms Nearest Neighbor (NNA) and Partitioning Around Medoids (PAM) clustering algorithms are considered for dividing the circuits into sub circuits. Clustering is alternatively referred to as unsupervised learning segmentation. The clusters are formed by finding the similarities between data according to characteristics found in the actual data. NNA is a serial algorithm in which the items are iteratively merged into the existing clusters that are closest. PAM represents a cluster by a medoid.

## 2. Circuit partitioning concept

VLSI circuit partitioning is a vital part of physical design stage. The essence of circuit partitioning is to divide the circuit into a number of sub-circuits with minimum

interconnections between them. This can be accomplished by recursively partitioning a circuit into two parts until we reach desired level of complexity. Thus two way partitioning is basic problem in circuit partitioning, which can be described as (Dutt& Deng, 1996).

Logic netlist can be represented as a hypergraph H (V,Eh) where

- Each node v∈V in hypergraph represents a logic cell of the netlist, and
- Each hyperedge e∈ Eh represents a net connecting various logic cells
- Various weights representing different attributes are attached to all nodes and edges of the hypergraph H, these are: On nodes: area estimates, On edges: length (after global placement)

The problem is:

- To partition the set V of all nodes v∈V into a set of disjoint subsets, of V, such that each node v is present in exactly one of these subsets. These subsets are referred to as blocks of the partition.
- The partition on V induces a cut of the set of all hyper edges, that is, Eh. A cut is subset of Eh, such that for every hyper edge h present in the cut there are at least two nodes adjacent to h, which belong to separate blocks of the partition.
- The objective function of partitioning approach has to address the following issues:
- It should be able to handle multi-million node graphs in a reasonable amount of computation time
- It should attempt to balance the area attribute of all the blocks of the partition with the additional constraint that there is an area penalty associated with every hyperedge that get cut.
- It should try to minimize interconnections between different clusters so as to satisfy the technological limit on the maximum number of interconnects allowed.

## 3. Memetic approach in VLSI circuit partitioning

A new approach Memetic Algorithm is described in this section to solve problem of circuit partitioning pertaining to VLSI.

### 3.1 A model to solve circuit partitioning

The circuit partitioning problem can be formally represented in graph theoretic notation as a weighted graph, with the components represented as nodes, and the wires connecting them as edges, the weights of the node represent the sizes of the corresponding components, and the weights of the edges represent the number of wires connecting the components. In its general form, the partitioning problem consists of dividing the nodes of the graph into two or more disjoint subsets such that the sum of weights of the nodes in each subset does not exceed a given capacity, and the sum of weights of edges connecting nodes in different subsets is minimized. But generally the circuits are represented as bipartite graphs consisting of two sets of nodes, the cells and the nets/ Edges connect each cell to several nets, and each net to several cells as shown in Fig1.Let G= (M, N, E), mi ∈ is a cell, ni∈N is a net, and eij=(mi,nj) ∈E is an edge which represents that mi and nj are connected electrically. For any nj for all I for which eij exists, we say that the cells mi are connected by net nj.

Conversely, for any mi for all j for which eij exists, we say that the nets nj are connected of cell mi. Each cell mi has an area ai, and each net nj has a cost cj. The edges of the bipartite graph are un weighted. In this case, the partitioning problem is to divide the set of cells into disjoint subsets, M1, M2,…….Mk, such that the sum of cell areas in each subset Mi is less than a given capacity Ai, and the sum of costs of nets connected to cells in different subsets is minimized. That is,

$$\bigcup_{n=1}^{k} M \quad \text{n} = \text{M}, \tag{1}$$

$\forall$Mn, $\Sigma\forall$mi$\varepsilon$Mnai $\leq$ An

and $\forall$ nj, if nj is connected to cells in p different partitions, then,

$$C = \Sigma(p\text{-}1)cj \text{ is minimized} \tag{2}$$

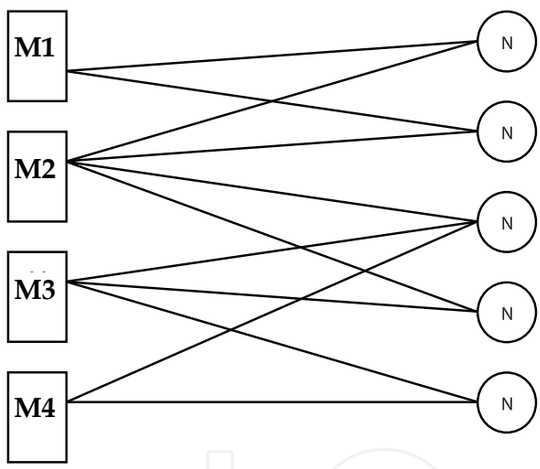
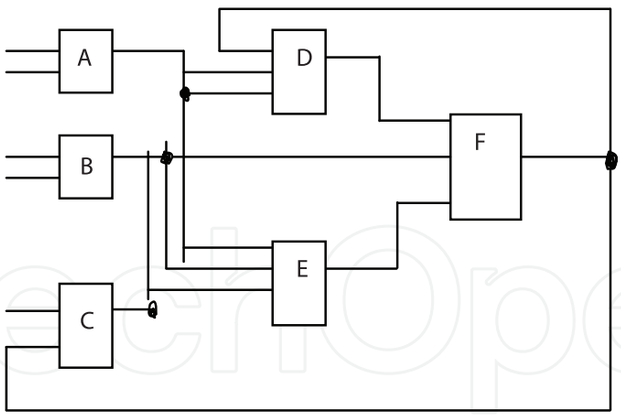



Fig. 1. Bipartite graph model for partitioning

### 3.2 Memetic algorithms applied to circuit partitioning

i. Chromosome Representation

1 bit in the chromosome represents each cell, the value of which determines the partition in which the cell is assigned (Krasnogor & Smith, 2008). The chromosome is sorted as an array of 32 bit packed binary words. The net list is traversed in a breadth-first search order, and the cells are assigned to the chromosome in this order. Thus, if two cells are directly connected to each other, there is a high probability that their partition bits are close to each other in the chromosome. An example is the breadth-first search sequence and the corresponding chromosome as shown in Fig. 2.




| Cell | Connected To | Bipartition |
|---|---|---|
| A | D,E | 1 |
| B | E,F | 1 |
| C | D,E,F | 1 |
| D | A,C,F | 0 |
| E | A,B,C,F | 0 |
| F | B,C,D,E | 0 |
| BFS Sequence | A D E C F B | |

Bipartitioning Chromosome | 1 0 0 1 0 1 |

Fig. 2. Breadth-first search sequence and the corresponding chromosome

ii. Fitness Scaling

Fitness scaling is used to scale the raw fitness values of the chromosomes so that the GA sees a reasonable amount of difference in the scaled fitness values of the best versus the worst individuals.

The following fitness algorithm applies to evaluation functions that determine the cost, rather than the fitness, of each individual (Univesity of New Mexico, 1995). From this cost, the fitness of each individual is determined by scaling as follows.

A referenced worst cost is determined by

$$Cw=C+S\sigma \tag{3}$$

Where C is the average cost of the population, S is the user defined sigma-scaling factor, and σ is the standard deviation of the cost of the population. In case Cw is less than the real worst cost in the population, they only the individuals with cost lower than Cw are allowed to participate in the crossover.

Then, the fitness of each individual is determined by

$$\begin{cases} \text{Cw-C} & \text{if Cw>0} \\ \text{F=0} & \text{otherwise} \end{cases} \tag{4}$$

This scales the fitness such that, if the cost is ±k standard deviations from the population average, the fitness is

$$F=(S \pm K)\sigma \tag{5}$$

This means that may individuals worse than S standard deviation from the population mean (k=s) are not selected at all. If S is small, the ratio of the lowest to the highest fitness in the population increases, and then the algorithm becomes more selective in choosing parents. On the other hand, if S is large, then Cw is large, and the fitness values of the members of the population are relatively close to each other. This causes the difference in selection probabilities to decrease and the algorithm to be less selective in choosing parents.

iii. Evaluation

The cut cost is calculated as the number of nets cuts. If the net is present in both partitions, or if the net is present in the partition opposite to its I/O pad, then it is said to have a cut (Merz & Freisleben, 2000).

Counting number of 1's in the chromosome does partition imbalance evaluation. A quadratic penalty has been used for imbalance, so that large imbalance is penalized more than a small imbalance. The user specifies the relative weights for cut and imbalance Wc and Wb.

Thus the total cost is:

$$C = Wc(\text{cut}) + Wb(\text{imbalance})2 \tag{6}$$

iv. Incorporation and Duplicate Check

The two new offspring formed in each generation are incorporated into the population only if they are better than the worst individuals of the existing population. Before entering a new offspring into the population, it is checked against all other members of the population having the same cost, in order to see whether it is duplicate. Duplicates can result due to the same crossover operation (T. Jones, 1995).

Duplicates have two disadvantages:

- First they occupy storage space that could otherwise be used to store a population with more diverse feature.
- Second whenever crossover occurs between two duplicates, the offspring is identical to the parents, regardless of the cut point, and this tends to fill the population with even more duplicates.

v. Mutation

After crossover and incorporation, mutation is performed on each bit of the population with a very small probability Pm. We go through the entire population once (Krasnogor et al., 1998a). For each mutation the location in bits is determined from previous location and a random number as follows,

$$\text{Next}=\text{Prev}+\left\lceil \frac{\text{Log}(\text{rand}[0,1])}{\text{Log}(1.0-\text{PM})} \right\rceil \quad (7)$$

Where PM is the mutation probability.

Each mutation is evaluated and accepted separately, and this process is continued until end of population is reached. The mutated version replaces the unmutated version of the same individual in the population. The acceptance of mutation operation has some probabilistic characteristics similar to simulated annealing. If the change in the cost C is negative, signifying that the fitness has increased, the mutation is always accepted, as in simulated annealing. If change in the cost is positive, then mutations are accepted probabilistically.

## 4. Evolutionary time series model for partitioning using Neuro-Memetic approach

An evolutionary time-series model for partitioning a circuit is discussed using Neuro Memetic algorithm owing to its local search capability.

**Sample Data Set**

A sample example and the corresponding chromosome representation is shown in Fig 3 and Fig 4.

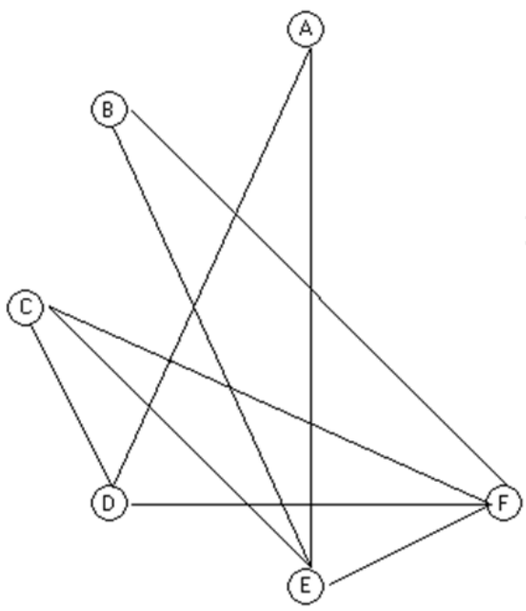



Fig. 3. Sample Circuit

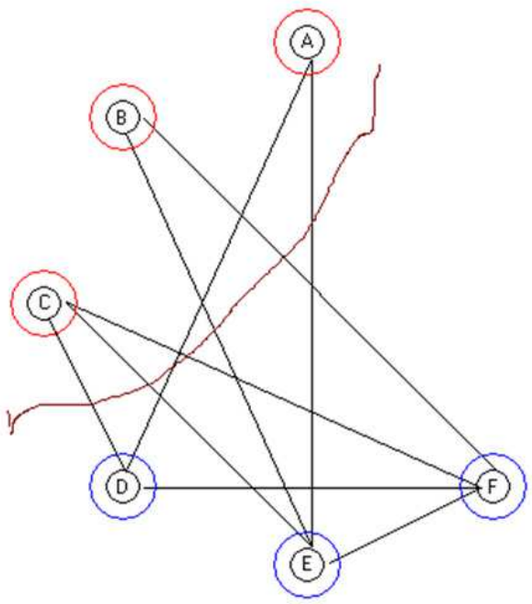



Fig. 4. Bipartition Circuit

Sub circuit 1 A, B, C total edges = 7

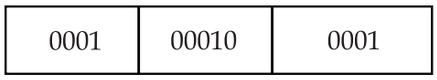
Sub circuit 2 D, E, F total edges =10

| Cell | No of edges | Bipartition |
|---|---|---|
| A | 2 | 1 |
| B | 2 | 1 |
| C | 3 | 1 |
| D | 3 | 0 |
| E | 3 | 0 |
| F | 4 | 0 |

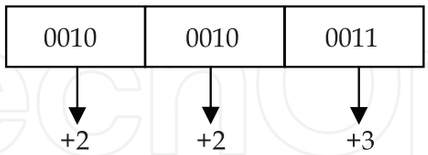
Chromosomes representation: Sub circuit1 (A, B, C) 0010 0010 0011, Sub circuit2 (D, E, F) 0011 0011 0100

- Neuro-Memetic Model: Neuro-memetic model makes it possible to predict the sub circuit from circuit with minimum interconnections between them.
- Training Procedure: The purpose of the training process is to adjust the input and output parameters of the NN (Neural Network) model, so that the MAPE (Mean Absolute Percentage Error) measure is minimized. Training of the feed-forward neural network models is usually performed using back propagation learning algorithms. Most often, the error surface becomes trapped to local minima, usually not meeting the desired convergence criterion. The termination at a local minimum is a serious problem while the neural network is learning. In other words, such a neural network is not completely trained (Oxford Univ Press, 1995). Another issue where care must be taken is "the receptiveness to over-fitting". But, memetic algorithms offer competent search method for intricate (that is, possessing many local optima) spaces to find nearly local optima. Thus, its ability to find a better suboptimal solution or have a higher probability to obtain the local optimal solution makes it one of the preferred candidates to solve the learning problem.
- Training with MA: The parameters of the neural network are tuned by a memetic algorithm (Krasnogor et al., 1998b) with arithmetic crossover and non uniform mutation. A population (P) with 200 genotypes is considered. They are randomly initialized, with maximum number of iterations fixed at 200 and MA is run for 100 generations with the same population size. The best model was found after 63 generations. In this method, the probability of crossover is 0.6 and the probability of mutation is 0.2. These probabilities are chosen by trial and error through experiments for good performance. The new population thus generated replaces the current population. The above procedures are repeated until a certain termination condition is satisfied. The number of the iterations required to train the MA-based neural network is 2000. The range of the fitness function of neural network is (0, 1).
- Evaluate individuals using the fitness function: The objective of the fitness function is to minimize the prediction error. In order to prevent over-fitting and to give more exploration to the system, the fitness evaluation framework is changed and use the weight imbalance to calculate the fitness of a chromosome. The fitness of a chromosome for the normal class is evaluated as shown in the example below.

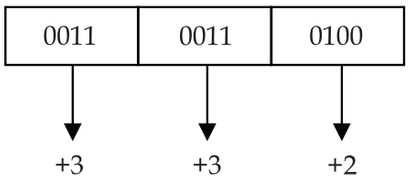
Take the testing samples




Now take the sub circuit 1 with data set (d1)




For sub circuit 2 data set d2




Calculate the sum of (+) credit & (-) debit for each sample data d1 & d2

For d1=+2+2+3=7

d2=+3+3+2=8 so it is found that **sample fitness of data d1** is best sample.

### 4.1 Design of the system to recognize sub circuit with minimum interconnection

The present task involves the development of Neural Network, which can train to recognize sub circuit with minimum interconnection between them, from a large circuit given. Following are the steps involved in design of the system

1. Create a input data file which consists of training pairs.
2. In data extraction, a circuit is bipartite and chromosomes are represented for each sub circuit.
3. Design the neural network based upon the requirement and availability.
4. Simulate the software for network.
5. Initialize count=0, fitness=0, number of cycles.
6. Generation of Initial Population. The chromosome of an individual is formulated as a sequence of consecutive genes, each one coding an input parameter.
7. Initialize the weight for network. Each weight should be set to a random value between –0.1 to 1.
8. Calculates activation of hidden nodes.

$$\text{xJh}=\frac{1}{1+e-(\sum wjkh)*\text{xpn}} \tag{8}$$

9. Calculate the output from output layers

$$\text{xio}=\frac{1}{1+e-(\sum wijo)*\text{xjh}} \tag{9}$$

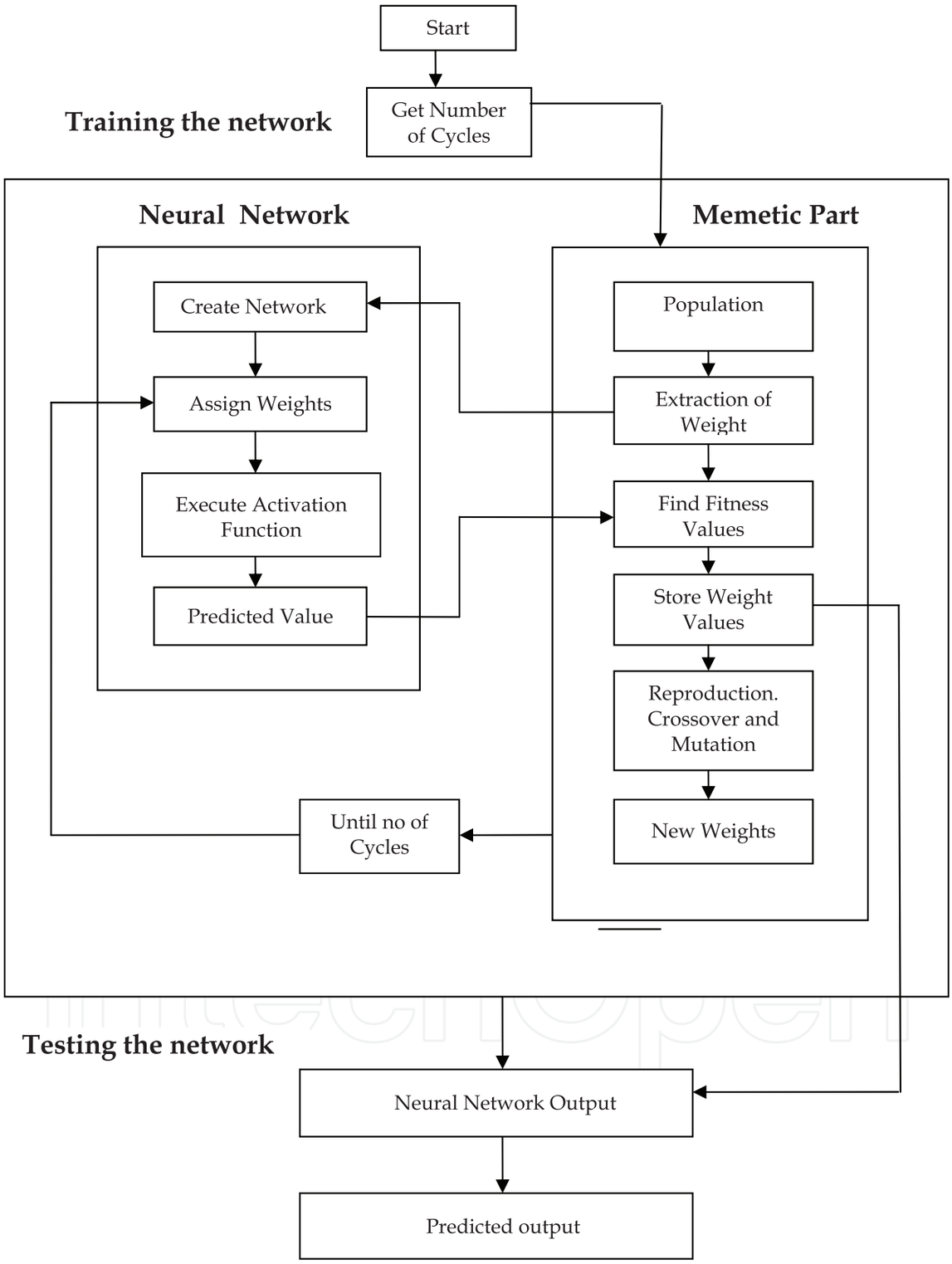



Fig. 5. Recognize Sub Circuit with Minimum Interconnection

10. Compares the actual output with the desired outputs and find a measure of error. The genotypes are evaluated on the basis of the fitness function.
11. If (previous fitness < current fitness value) then store current weights.
12. Count = Count + 1
13. Selection: Two parents are selected by using the Roulette wheel mechanism.
14. Genetic Operations: Crossover, Mutation and Reproduction to generate new weights (Apply new weights to each link).
15. If (number of cycles> count) Go to Step 7
16. training set is reduced to an acceptable value.
17. Verify the capability of neural network in recognition of sub circuit with minimum interconnection between them.
18. End.

- Development of Neural Network: In the context of recognition of sub circuit with minimum interconnection, the 3-layer neural network is employed to learn the input-output relationship using the MA. The layers of input neuron are responsible for inputting. The number of neurons in this output layer is determined by the size of set of desired output, with each possible output being represented by separate neuron. Neural network contains 12 input nodes, 20 neurons in the first hidden layer, 14 neurons in the second hidden layer and the output layer has 2 neurons. It results in a 12-14-2 Back propagation neural network. Sigmoid function is used as the activation function. Memetic Algorithm is employed for learning (Holstein & Moscato, 1999). For the back-propagation with momentum and adaptive learning rate, the learning rate is 0.2, the momentum constant is 0.9. During the training process the performance of 0.00156323 was obtained at 2000 epochs.

## 5. Neuro–EM and neuro-k mean clustering approach for VLSI design partitioning

This section is focused in use of clustering methods k-means (J. B. MacQueen, 1967) and Expectation-Maximization (EM) methodology (Kaban & Girolami, 2000).

### 5.1 Neuro-EM model

The system consists of three parts each dealing with data extraction, Learning stage and recognition stage. In data extraction, a circuit is bipartite and partitions it into 10 clusters, a user-defined value, by using K-means (J. B. MacQueen, 1967) and EM methodology (Kaban & Girolami, 2000), respectively. In recognition stage the parameters, that is, centroid and probability are fed into generalized delta rule algorithm separately and train the network to recognize sub-circuits with lowest amount of interconnections between them. Block diagram of model to recognize sub-circuits with lowest amount of interconnections between them using two techniques K-means and EM methodology with neural network are shown in Fig.6 and Fig.7.

In recognition stage the parameters, that is, centroid and probability are fed into generalized delta rule algorithm separately and train the network to recognize sub circuit with minimum interconnection between them. Block diagram of model for Partitioning a Circuit are depicted in Fig. 8.

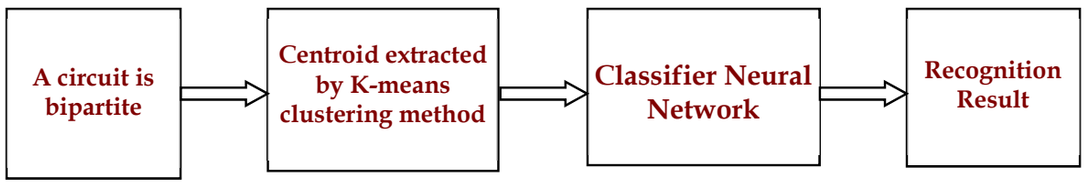
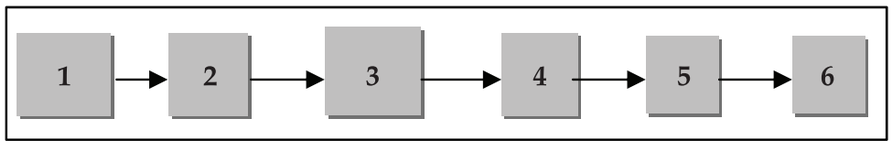



Fig. 6. Block diagram of K-means with neural network




Fig. 7. Block diagram of EM methodology with neural network




1. Circuit is bipartite and data represented
2. Applying K-means
3. Applying EM methodology
4. centroid and probability
5. Neural network
6. Recognition Result

Fig. 8. Block Diagram of Model for Partitioning a Circuit

### 5.2 Sample data set

A sample example representation is shown in Fig.9 and Fig 10

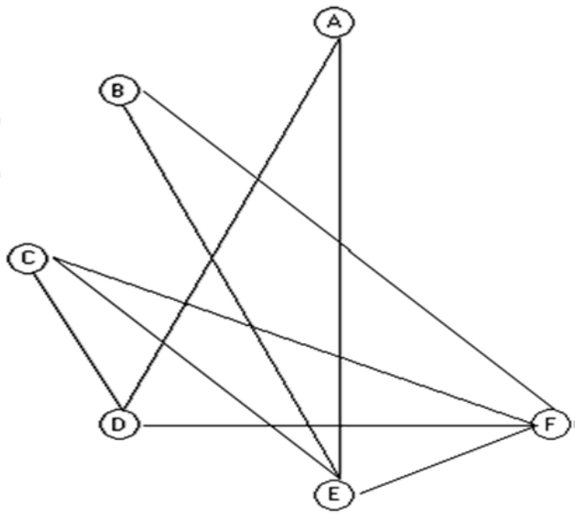



Fig. 9. Sample Circuit

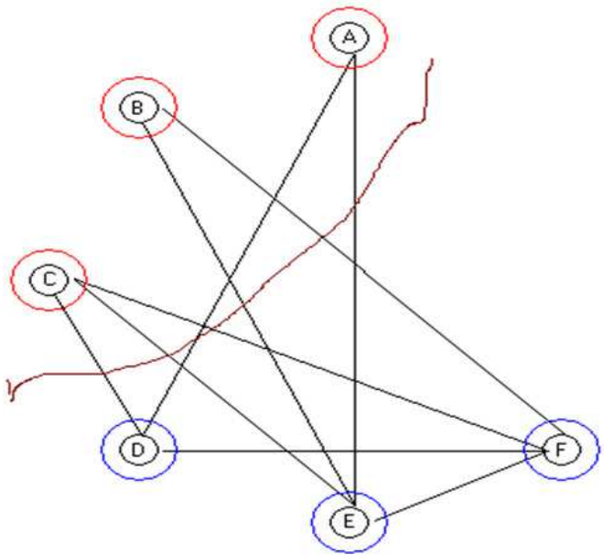



Fig. 10. Bipartition Circuit
Sub circuit 1 A, B, C total edges = 7, Sub circuit 2 D, E, F total edges =10

| Cell | No of edges | Bipartition |
|---|---|---|
| A | 2 | 1 |
| B | 2 | 1 |
| C | 3 | 1 |
| D | 3 | 0 |
| E | 3 | 0 |
| F | 4 | 0 |

Table 1. Bipartation Matrix
Data representation: Sub circuit1 (A, B, C) 0010 0010 0011,
Sub circuit2 (D, E, F) 0011 0011 0100

### 5.3 Expectation Maximization algorithms

The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin in the Journal of the Royal Statistical Society (Arthur et al.,1997). They pointed out that method had been "proposed many times in special circumstances" by other authors, but the 1977 paper generalized the method and developed the theory behind it.

The EM algorithm for clustering is described in detail in Witten and Frank (2001) (Witten & Frank, 2005). The Expectation-Maximization (EM) algorithm is part of the Weka clustering package. EM is a statistical model that makes use of the finite Gaussian mixtures model. The basic approach and logic of this clustering method is as follows. Suppose a single continuous variable in a large sample of observations is measured. Further, suppose that the sample consists of two clusters of observations with different means (and perhaps different standard deviations) within each sample, the distribution of values for the continuous variable follows the normal distribution. The resulting distribution of values (in the population) may look as shown in Fig.11.

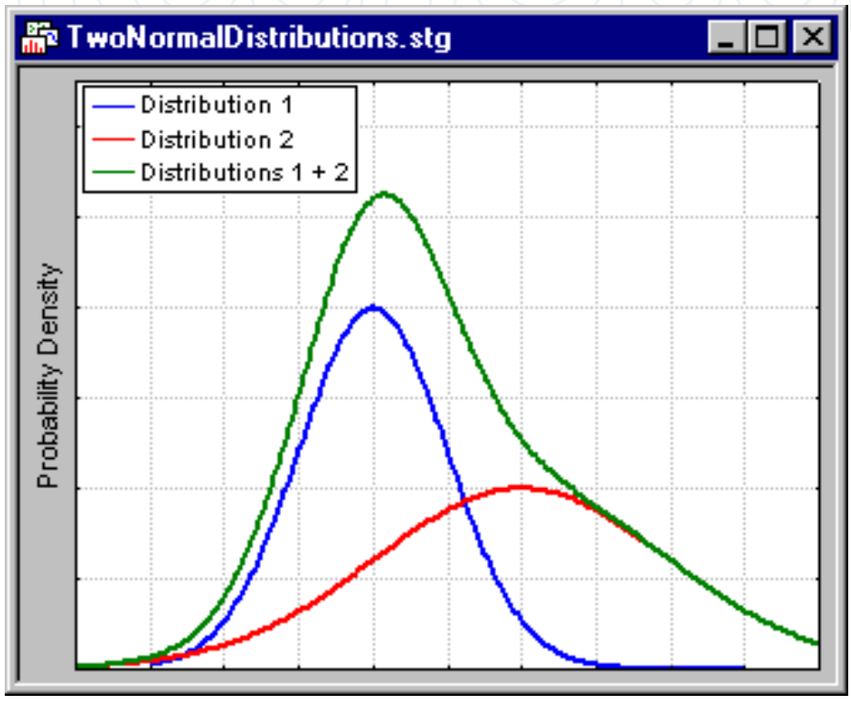



Fig. 11. Two normal distributions of EM Algorithm (Screen Shot)

i. Mixtures of distributions. The illustration in Fig 5.12 shows two normal distributions with different means and different standard deviations and the sum of the two distributions. Only the mixture (sum) of the two normal distributions (with different means and standard deviations) would be observed. The goal of EM clustering is to estimate the means and standard deviations for each cluster so as to maximize the likelihood of the observed data (distribution). Put another way, the EM algorithm attempts to approximate the observed distributions of values based on mixtures of different distributions in different clusters.
With the implementation of the EM algorithm in some computer programs, one may be able to select (for continuous variables) different distributions such as the normal, log-normal, and Poisson distributions (Karlis, 2003) and can select different distributions for different variables, thus derive clusters for mixtures of different types of distributions.

ii. Categorical variables. The EM algorithm can also accommodate categorical variables. The method will at first randomly assign different probabilities (weights, to be precise) to each class or category, for each cluster. In successive iterations, these probabilities are

refined (adjusted) to maximize the likelihood of the data given the specified number of clusters (Kim, 2002).

iii. Classification probabilities instead of classifications. The results of EM clustering are different from those computed by k-means clustering. The latter will assign observations to clusters to maximize the distances between clusters. The EM algorithm does not compute actual assignments of observations to clusters, but classification probabilities. In other words, each observation belongs to each cluster with a certain probability. Of course, as a final result one can usually review an actual assignment of observations to clusters, based on the (largest) classification probability (Gyllenberg et al., 2000).

The algorithm is similar to the K-means procedure in that a set of parameters are recomputed until a desired convergence value is achieved. The parameters are recomputed until a desired convergence value is achieved. The finite mixtures model assumes all attributes to be independent random variables.

A mixture is a set of *N* probability distributions where each distribution represents a cluster. An individual instance is assigned a probability that it would have a certain set of attribute values given it was a member of a specific cluster. In the simplest case N=2 the probability distributes are assumed to be normal and data instances consist of a single real-valued attribute. Using the scenario, the job of the algorithm is to determine the value of five parameters specifically,

1. The mean and standard deviation for cluster 1
2. The mean and standard deviation for cluster 2
3. The sampling probability *P* for cluster 1 (the probability for cluster 2 is 1-*P*)

the general procedure is given below,

1. Guess initial values for the five parameters.
2. Use the probability density function for a normal distribution to compute the cluster probability for each instance. In the case of a single independent variable with mean μ and standard deviation σ, the formula is:

$$f(x) = \frac{1}{(\sqrt{2\pi}\sigma)e^{\frac{-(?-\mu)^2}{2\sigma^2}}} \tag{10}$$

In the two-cluster case, there are two probability distribution formulae each having differing mean and standard deviation values.

1. Use the probability scores to re-estimate the five parameters.
2. Return to Step 2

The algorithm terminates when a formula that measures cluster quality no longer shows significant increases. One measure of cluster quality is the likelihood that the data came from the dataset determined by the clustering. The likelihood computation is simply the multiplication of the sum of the probabilities for each of the instances. With two clusters *A* and *B* containing instances x1, x2, … xn where PA = PB = 0.5 the computation is:

$$\left[.5P(x_1 \mid A) + .5P(x_1 \mid B)\right]\left[.5P(x_2 \mid A) + .5P(x_2 \mid B)\right]...\left[.5P(x_n \mid A) + .5P(x_n \mid B)\right] \tag{11}$$

Algorithm is similar to K-mean procedure, in that sets of parameters are re-computed until desired convergence value is achieved. General procedure is

- Initialize parameters.
- Use the probability density function for normal distribution to compute cluster probability for each instance. For example in the case of two-cluster one will have the two probability distribution formulae each having different mean and standard deviation values.
- Use the probability scores to re-estimate the parameter.
- Return to step 2.
- The algorithm terminates when formula that measure cluster quality exists no longer.

The tool shed output of this algorithm would be the probability for each cluster. EM assigns a probability distribution to each instance, which indicates the probability of it belonging to each of the clusters.

In the context of recognizing the sub circuit from circuit with minimum interconnections between them, artificial neurons is structured into three normal types of layers input, hidden and output which can create artificial neural networks. The layers of input neuron are responsible for inputting a feature vectors that is, centroid and probability, which are extracted from K-means and EM algorithms respectively. The number of neurons in this output layer is determined by size of set of desired output, with each possible output being represented by separate neuron. Between these two layers there can be many hidden layers. These internal layers contain many of the neuron in various interconnected structures.

### 5.4 Design of the system to recognize sub circuit with minimum interconnections

The present task involves the development of neural network, which can train to recognize sub circuit with minimum interconnection between them from large circuit given.

Following are the steps involved in design of the system,

1. Create a input data file which consists of training pairs.
2. In data extraction, a circuit is bipartite and data are represented for each sub circuit.
3. Centroid and probability features are extracted from K-means and EM algorithms
4. Design the neural network based upon the requirement and availability.
5. Simulate the software for network.
6. Train the network using input data files until error falls below the tolerance level.
7. Verify the capability of neural network in recognition of test data

Algorithm:

The learning algorithm of back propagation network is given by “generalized delta rule”.

Step 1. The algorithm takes input vector (features) to the back propagation network.

Step 2. let K be number of nodes in the layer determined by length of training vectors that is number of feature N. Let j be number of nodes in hidden layer. Let I be number of nodes in output layer. Denote activation of hidden layer as xjh and in output layer is xio. Weight connecting input layer and hidden layer are wjkh and weight connecting hidden layer and output layer is wijo.

Step 3. Initialize the weight for network. Each weight should be set to a random value between –0.1 to 1.

Step 4. Calculates activation of hidden nodes

$$\text{xJh}=g(\sum wjkh * \text{xpn})=\frac{1}{1+e-(\sum wjkh * \text{xpn})} \tag{12}$$

Step 5. Calculate the output from output layers

$$\text{xio}=g(\sum wjko * \text{xjh})=\frac{1}{1+e-(\sum wijo * \text{xjh})} \tag{13}$$

Step 6. Compares the actual output with desired outputs and finds a measure of error.

Step 7. After comparison it finds in which direction (+ or -) to change each weight in order to reduce error.

Step 8. Find the amount by which to change each weight. It applies the corrections to the weight and repeat all above steps with all training vectors until the error for all the vectors in training set is reduced to an acceptable value.

Step 9: End.

## 6. Evaluation of fuzzy ARTMAP with DBSCAN in VLSI partition application

This section describes a new model for partitioning a circuit using DBSCAN and fuzzy ARTMAP neural network.

### 6.1 Overview of art map

The basic ART system is an unsupervised learning model. It typically consists of a comparison field and a recognition field composed of neurons, a vigilance parameter, and a reset module. The vigilance parameter has considerable influence on the system, higher vigilance produces highly detailed memories (many, fine-grained categories), while lower vigilance results in more general memories (fewer, more-general categories). The comparison field takes an input vector (a one-dimensional array of values) and transfers it to its best match in the recognition field. Its best match is the single neuron whose set of weights (weight vector) most closely matches the input vector. Each recognition field neuron outputs a negative signal (proportional to that neuron's quality of match to the input vector) to each of the other recognition field neurons and inhibits their output accordingly. In this way the recognition field exhibits lateral inhibition, allowing each neuron in it, to represent a category to which input vectors they are classified. After the input vector is classified, the reset module compares the strength of the recognition match to the vigilance parameter. If the vigilance threshold is met, training commences. Otherwise, if the match level does not meet the vigilance parameter, the firing recognition neuron is inhibited until a new input vector is applied. The training commences only upon completion of a search procedure. In the search procedure, recognition neurons are disabled one by one, by the reset function until the vigilance parameter is satisfied by a recognition match. If no

committed recognition neuron's match meets the vigilance threshold, then an uncommitted neuron is committed and adjusted towards matching the input vector.

There are two basic methods of training ART-based neural networks: slow and fast. In the slow learning method, the degree of training of the recognition neuron's weights towards the input vector is calculated to continuous values with differential equations and is thus dependent on the length of time the input vector is presented. The basic structure of the ART based neural network is shown in Fig 12 With fast learning, algebraic equations are used to calculate degree of weight adjustments to be made, and binary values are used. While fast learning is effective and efficient for a variety of tasks, the slow learning method is more biologically plausible and can be used with continuous-time networks (that is, when the input vector can vary continuously). Fig 13 shows the fast learning ART-based neural network.

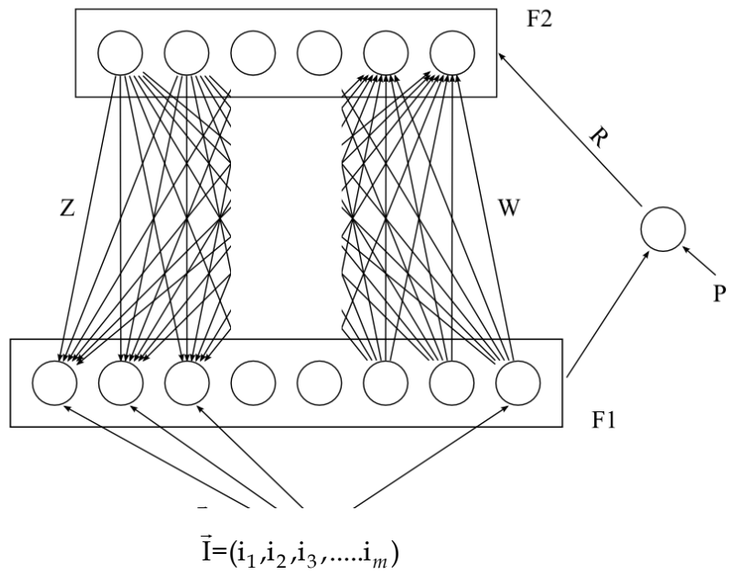



Fig. 12. Basic ART Structure

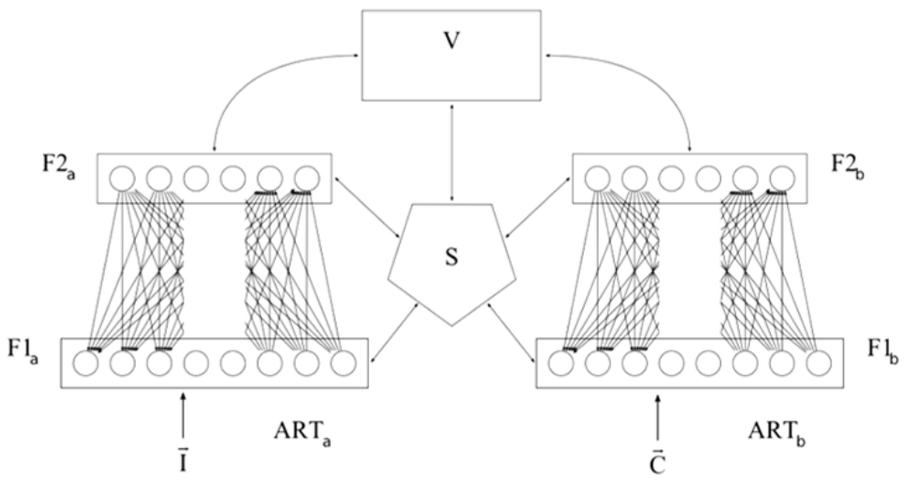



Fig. 13. Fast learning ART-based neural network

The first principle of Adaptive Resonance Theory (ART) was first introduced by Grossberg in 1976 (Carpenter,1997), whose structure resembles those of feed-forward networks. The simplest variety of ART networks is accepting only binary inputs which is called as ART (Grossberg,1987,2003). It was then extended for network capabilities to support continuous inputs called as ART-2 (Carpenter & Grossberg ,1987).ARTMAP (Carpenter et al.,1987),also known as Predictive ART, combines two slightly modified ART-1 or ART-2 units into a supervised learning structure where the first unit takes the input data and the second unit takes the correct output data and then used to make the minimum possible adjustment of the vigilance parameter in the first unit in order to make the correct classification.

### 6.2 Fuzzy ARTMAP

Fuzzy logic with the combination of Adaptive Resonance Theory gives Fuzzy ARTMAP, is a class of neural network that perform supervised training of recognition pattern and maps in response to input vectors generated. Fuzzy ART (Carpenter et al.,1991) implements fuzzy logic into ART's pattern recognition, thus enhancing generalizability. An optional (and very useful) feature of fuzzy ART is complement coding, a means of incorporating the absence of features into pattern classifications, which goes a long way towards preventing inefficient and unnecessary category proliferation. The performance of fuzzy ARTMAP depends on a set of user-defined hyper-parameters, and these parameters should normally be fine-tuned to each specific problem (Carpenter et al.,1992). The influence of hyper-parameter values is rarely addressed in ARTMAP literature. Moreover, the few techniques that are found in the literature for automated hyper-parameter optimization,example(Canuto et al., 2000; Dubrawski, 1997; Gamba & DellAcqua, 2003; C. Lim,1999), focus mostly on the vigilance parameter, even though there are four inter-dependent parameters (vigilance, learning, choice, and match tracking). A popular choice consists in setting hyperparameter values such that network resources (the number of internal category neurons, the number of training epochs, etc.) are minimized (Carpenter,1997). This choice of parameters may however lead to overtraining and significantly degrade the network. An effective supervised learning strategy could involve co-jointly optimizing both network (weights and architecture) and all its hyper-parameter values for a given problem, based on a consistent performance objective. Fuzzy ARTMAP neural networks are known to suffer from overtraining or over fitting, which is directly connected to a category proliferation problem. Overtraining generally occurs when a neural network has learned not only the basic mapping associated training subset patterns, but also the subtle nuances and even the errors specific to the training subset. If too much learning occurs, the network tends to memorize the training subset and loses its ability to generalize on unknown patterns. The impact of overtraining on fuzzy ARTMAP performance is two fold that is, an increase in the generalization error and in the resources requirements.

### 6.3 DBSCAN (Density-Based Spatial Clustering Of Applications with Noise)

DBSCAN is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaovei Xui in 1996(Ester, 1996). It is a density based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes. DBSCAN is one of the most common clustering algorithms and also

most cited in scientific literature. The basic DBSCAN algorithm has been used as a base for many other developments.

The overall structure of model is illustrated in Fig14 and Fig 15, Fig 16 show a sample circuit bipartite with related data set used . The feature extractor obtains feature vector for subcircuit, and is sent to training or inference module. The SFAM (simplified fuzzy ARTMAP) (Carpenter,1997) has two modules, that is, training and inference module. The feature vector of training subcircuits and the categories to which they belongs are specified to SFAM's training module. Once the training phase is complete, the vector represents the subcircuit with minimum interconnection. The test subcircuit pattern which is to be recognized with minimum interconnection is fed to inference module. Classifications of sub circuits are done by associating the feature vector with the top-down weight vectors (Carpenter et l., 1992;Caudell et al., 1994) in SFAM. The system can handle both symmetric and asymmetric circuit. In symmetric pattern, only distinct portion of circuit is trained whereas in asymmetric (1/2n)th portion of circuit is considered.

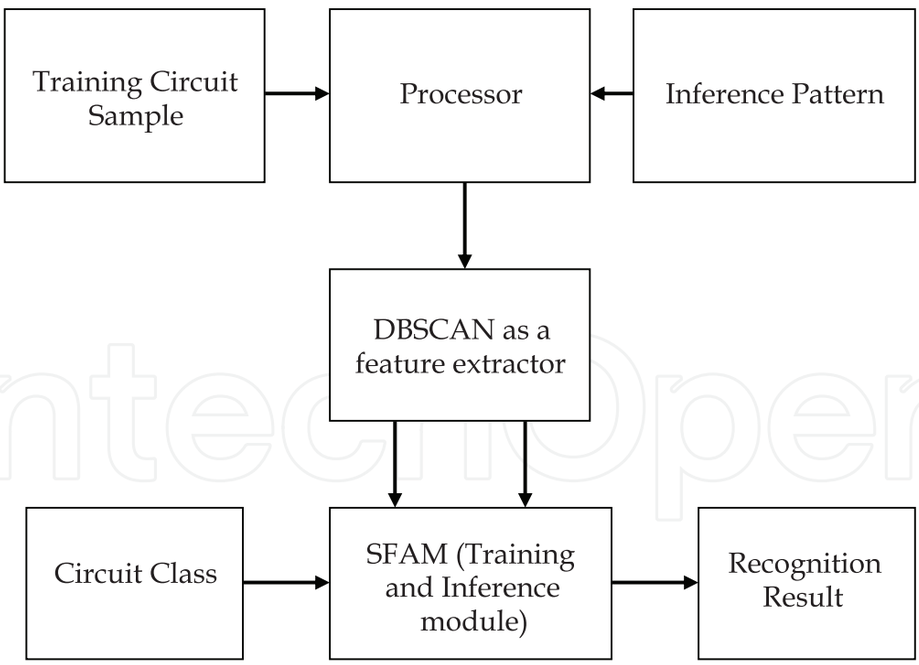



Fig. 14. Block diagram of recognition module for partitioning in VLSI Design

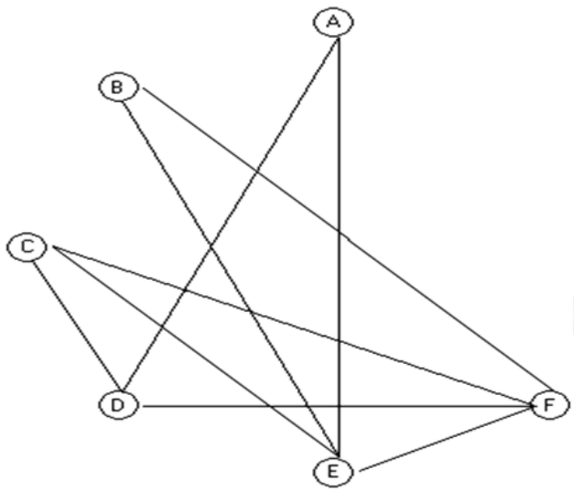



Fig. 15. Sample Circuit

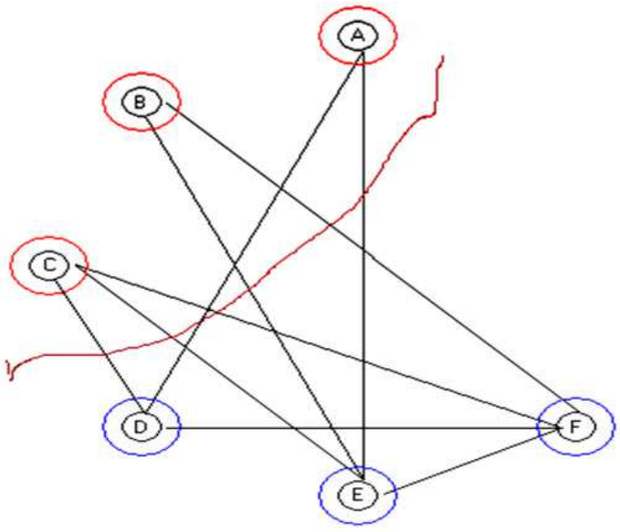



Fig. 16. Sample Bi-parted Circuit with data
Sample circuit bi parted Sub circuit 1 A, B, C total edges = 7 Sub circuit 2 D, E, F total edges =10

| Cell | No of edges | Bipartition |
|---|---|---|
| A | 2 | 1 |
| B | 2 | 1 |
| C | 3 | 1 |
| D | 3 | 0 |
| E | 3 | 0 |
| F | 4 | 0 |

Table 2. Bipartition Matrix
Data representation: Sub circuit1 (A, B, C) 0010 0010 0011,
Sub circuit2 (D, E, F) 0011 0011 0100

### 6.4 Overview of DBSCAN algorithm as a feature exactor

DBSCAN and clustering algorithm is used for feature Exactor which works on the densities (International Workshop on Text- Based information Retrieval (TIR 05),University of Koblenz-Landau, Germany). It separates the set D into subsets of similar densities. In the best case they can find out the cluster number k routinely and categorize the clusters of random shape and size.The runtime of this algorithms is in magnitude of O(n log(n)) for low-dimensional data (Busch,2005). A density-based cluster algorithm is based on two properties given below (TIR 05,University of Koblenz-Landau, Germany).

1. One is to define a region $C \subseteq D$, which forms the basis for density analyses.
2. Another is to propagate density information (the provisional cluster label) of C.

In DBSCAN a region is defined as the set of points that lie in the $\in$-neighborhood of some point p. if |C| exceeds a given Min Points-threshold Cluster label propagates from p to the other points in C. The complete description of DBSCAN algorithm is provided in (Ester et al., 1996;Tan et al., 2004;Dagher. I et al., 1999).

### 6.5 Simplified fuzzy ARTMAP module

In context of the circuit partitining in VLSI design to recognize the subcircuit with minimum interconnection between them, the size of input layer is 4 and output layer is 10. Hence it outcomes in 2-10 layered Fuzzy ARTMAP model.

Match and choice function for fuzzy ARTMAP in context to circuit partitioning is defined by,

For input vector I and cluster j from DBSCAN algorithm, Choice function given by

$$CFj(I)=\frac{|I \wedge Wj|}{\alpha+|Wj|} \tag{14}$$

Where $\alpha$ is small constant about 0.0000001,Wj is top-down weight

Winner node is one with highest activation /choice function, that is,

$$Winner=\max(CFj) \tag{15}$$

Match function which is very much used to find out whether the network must adjust its learning parameters is given by,

$$Fj(I)=\frac{|I \wedge Wj|}{|I|} \tag{16}$$

If MF j (I) $\geq$ vigilance parameter ($\rho$) then Network is in state of resonance, where $\rho$ is in range $0 \leq \rho \leq 1$.

If MF j (I) $\leq$ vigilance parameter ($\rho$) then Network is in state of mismatch reset.

## 7. A new clustering approach for VLSI circuit partitioning

The vital problem in VLSI for physical design algorithm is circuit partitioning. In this section concentration is on improving the partitioning technique using data mining approach. This

section deals with a range of partitioning methodological aspects which predicts to divide the circuit into sub circuits with minimum interconnections between them. This approach considers two clustering algorithms proposed by ( Li & Behjat, 2006) Nearest Neighbor(NN) and Partitioning Around Mediods(PAM) clustering algorithm for dividing the circuits into sub circuits. The experimental results show that PAM clustering algorithm yields better subcircuits than Nearest Neighbour. The experimental results are compared using benchmark data provided by MCNC standard cell placement bench netlists.

### 7.1 Considerations in choosing the right algorithm

Data mining algorithms have to be adapted to work on very large databases. Data reside on hard disks because they are too large to fit in main memory, therefore, algorithms have to make as few passes as possible over the data, as secondary memory fetch cycle increases the computational time and therefore reduces the run time performance. Quadratic algorithms are too expensive, that is the execution time of the operations in clustering algorithms is quadratic and so it becomes an important constraint in choosing an algorithm for the problem at hand. The aim in the thesis is to reduce the interconnections between the circuits with minimum amount of error,hence prototype based clustering is used. The attributes in the data set were less important, so the proximity matrix was created. Since both PAM and NNA belong to partitional and prototype based clustering and also the intention was to get the partition with the minimum interconnections these two algorithms were used.

### 7.2 Implementation

The implementation consists of three stages consisting of data extraction, partitioning and result using VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language) as a tool. In data extraction, a VLSI circuit represented as a bipartite graph is considered. The bipartite graph considered for the approach is shown in Fig 17.

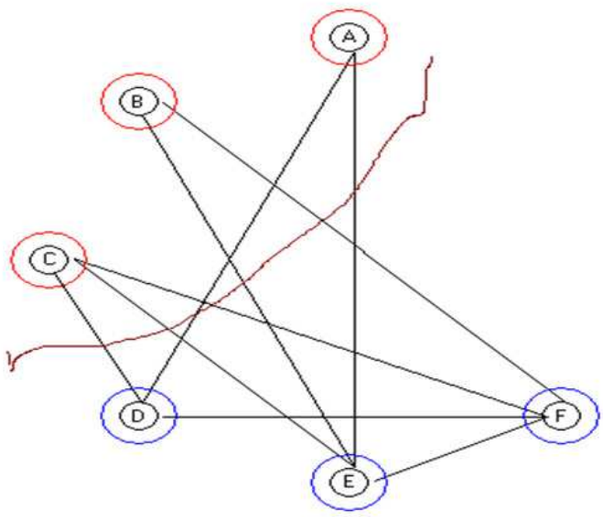



Fig. 17. Bipartition Circuit

Sub circuit 1 A, B, C total edges = 7

Sub circuit 2 D, E, F total edges =10

| Cell | No of edges | Bipartition |
|---|---|---|
| A | 2 | 1 |
| B | 2 | 1 |
| C | 3 | 1 |
| D | 3 | 0 |
| E | 3 | 0 |
| F | 4 | 0 |

Table 3. Bipartition Matrix

The block diagram to recognize sub-circuits with minimum interconnections using two techniques(Nearest Neighbor , PAM ).A new clustering algorithm is explored.

### 7.3 Applying clustering techniques to VLSI circuit partitioning

In adapting the two cluster partitioning algorithms to the area of VLSI circuit partitioning, the following considerations are of utmost importance.

The two algorithms take as input an adjacency matrix, which gives an idea of the similarity measure in the form of distances between the various data that are to be clustered. This approach uses this tool to partition circuits, so the circuit to be partitioned is the effective data to be clustered and the basic unit on which the algorithms will act are the nodes in a circuit.

Similarity between nodes in a circuit

Here, the input is the adjacency matrix, which defines the similarity between different nodes in the circuit. The attributes of nodes that are to be quantified as similarity between different nodes are based on several characteristics of logic gates such as,

1. Interconnections between nodes
2. Common signals as input
3. Functionality
4. Physical distance
5. Presence of the node on the maximum delay path

For example, if two nodes are interconnected, then the similarity between them is increased and the distance between them is reduced compared to two nodes which are not connected together.

Also, if some nodes get a common signal, such as a set of flip-flops sharing a common clock signal, it is desirable to have them partitioned into the same sub-circuit so as to reduce problems due to signal delay of synchronous control inputs. So, the distances between such nodes are also low.

The distance of a node to itself is taken as 0 and a low value of distance means the highest similarity. A high value of distance means maximum dissimilarity, and therefore least similarity, such nodes can be placed in different sub-circuits.

This adjacency or distance matrix is acted upon by the two algorithms, to effectively divide the circuit into sub-circuits, with the objective that is minimum interconnection under check. Adapting and applying data mining tools to VLSI circuit partitioning is a new approach. Improvisations and optimizations to the two algorithms are necessary and is essential to make them workable and viable as CAD tools.

Circuit chosen for implementation and testing

The circuit on which the two data mining algorithms are implemented (NNA and PAM) is as shown below. The circuit is a Binary Coded Decimal (BCD) code to seven segment code converter (Fig18). It has 4 inputs and 7 outputs. In this figure each rectangular block is considered as a node. A node is one which performs a defined function (Fig 19), it may be a simple AND gate or it may contain many interconnected flip-flops. So, a node contains one or more components and performs a logical function, the level of abstraction of a node can be changed to suit the basic unit understandable by a CAD tool.

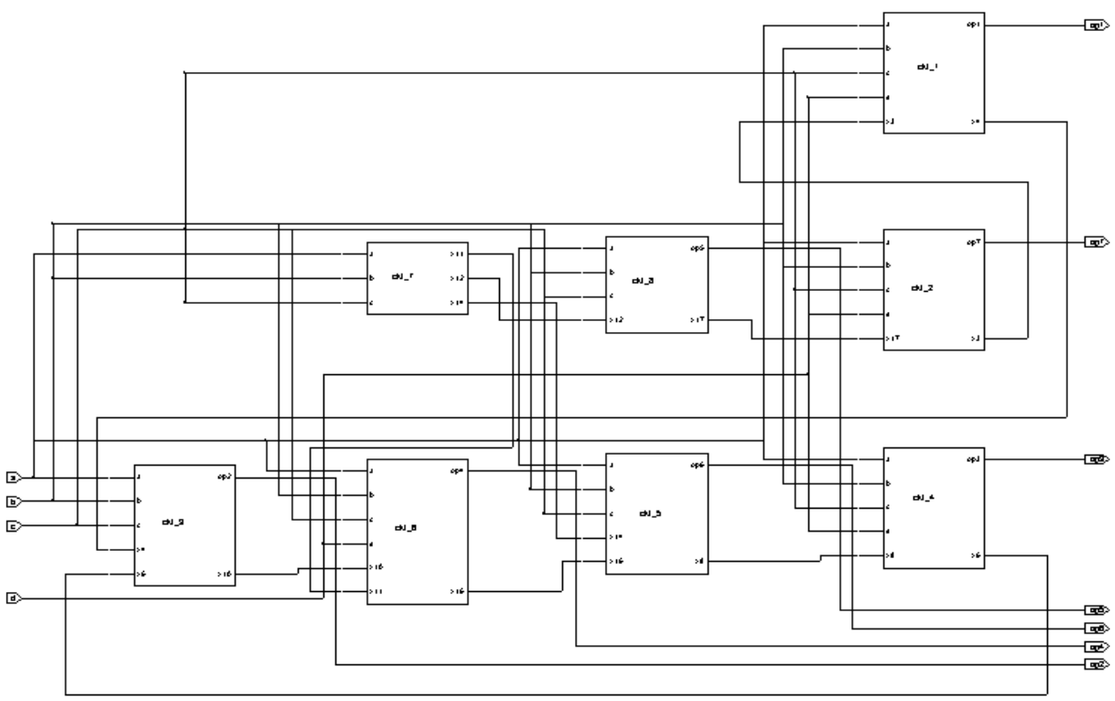


Fig. 18. Circuit of BCD code to Seven Segment code converter

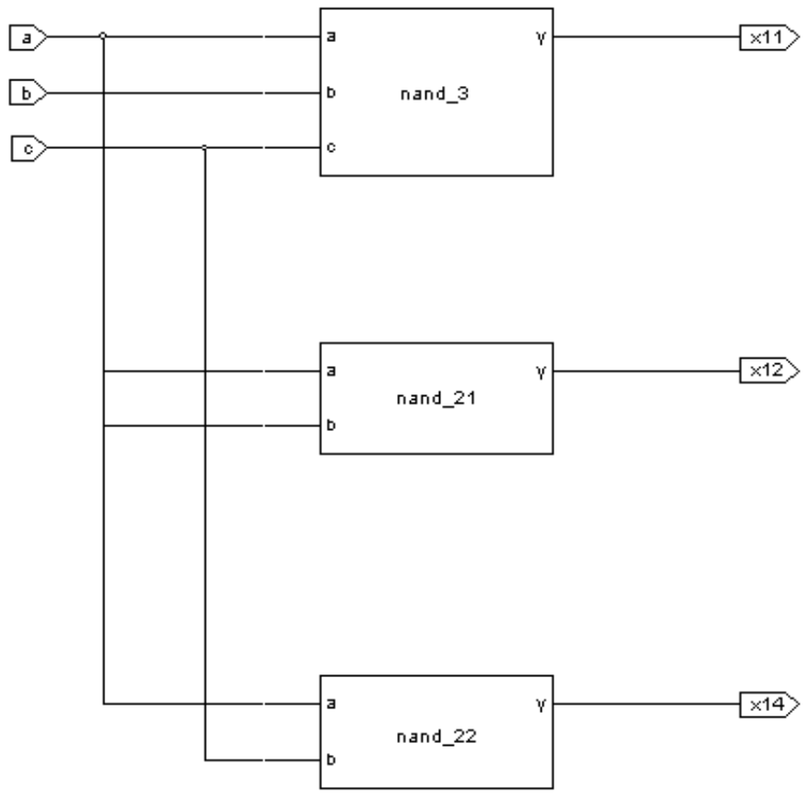



Fig. 19. A node (node 5) enlarged.

This shows that a node which is part of the main circuit consists of gates, such as Nand gate and or gates, or one which performs a logical function.

### 7.4 How to choose k and threshold value

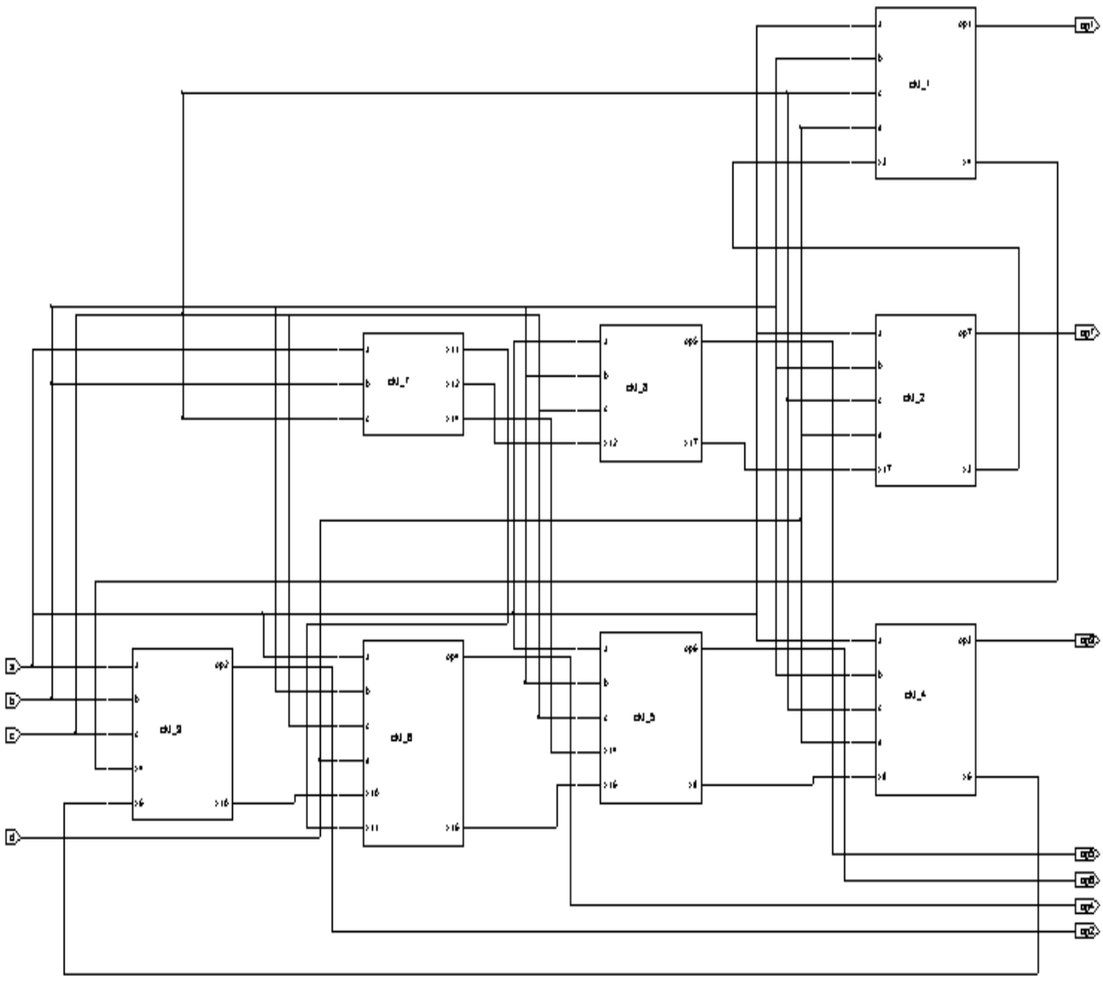
#### 7.4.1 PAM Algorithm – Choosing initial medoids

PAM starts from an initial set of medoids, by finding representative objects, called medoids, in clusters and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering. The PAM algorithm is based on the search for k medoids which are representative of the sequences based on the distance matrix. These k values should represent the structure of the sequences. After defining the set of k medoids, they would be used to construct the k clusters and partition the nodes by assigning each observation to the nearest medoid. In doing this, the target would be to identify the medoids that minimize the sum of the dissimilarities in the observations. As it can be seen, the choice of the initial medoids is very important. Medoid is the most centrally located point in a cluster, as a representative point of the cluster. The initial medoids chosen decides the quality of the formed clusters and the computational speed. If the initial medoids chosen are close to the final optimal medoids, yielding the final clusters with reduced cost, the computational cost will be reduced. Otherwise the number of iterations to find the final medoids will increase, this in turn increasing the time taken to obtain results and computational cost.

The representation by k-medoids has two advantages. First, it presents no limitations on attributes types and second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster, therefore it is less sensitive to the presence of outliers. Therefore, PAM is iterative optimization that combines relocation of points between perspective clusters with re nominating the points as potential medoids.Earlier the task is done to find out the optimum value of threshold "t", which decides the cluster density and quality, shows that the value of threshold from 2 to 5 gives optimal minimization of interconnections between sub-circuits. Therefore, for the two algorithms NNA and PAM, the threshold value of 2 and 3 are respectively chosen based on this task.

### 7.4.2 Details of the partitioned Circuits - Results on a Circuit with 8 Nodes is discussed

Fig.20 is an example of a Testing Circuit 1 with 8 nodes before applying the partitioning and the circuits after partitioning using the NN algorithm and applying the PAM algorithms are shown in Fig. 21. and Fig. 22. respectively.



.

Fig. 20. Circuit before applying partitioning techniques

The circuit shown in Fig7.10 is a BCD to seven-segment code converter before applying the partitioning algorithms and it has 8 nodes as shown in Fig 7.10. This circuit is tested in hardware and the functionality is concluded to be correct.

IntechOpen

Partitioned circuit obtained after applying Nearest Neighbor Algorithm

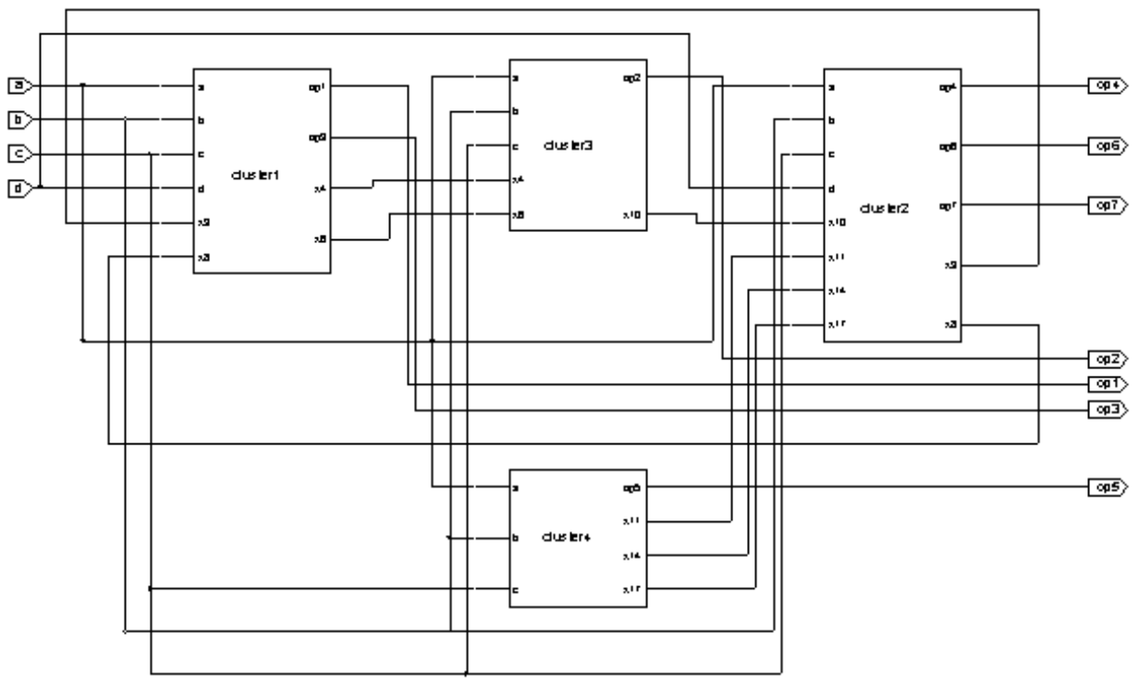



Fig. 21. NNA Partitioned circuit showing 4 sub-circuits

Partitioned circuit obtained after applying Partitioning Around Medoids algorithm

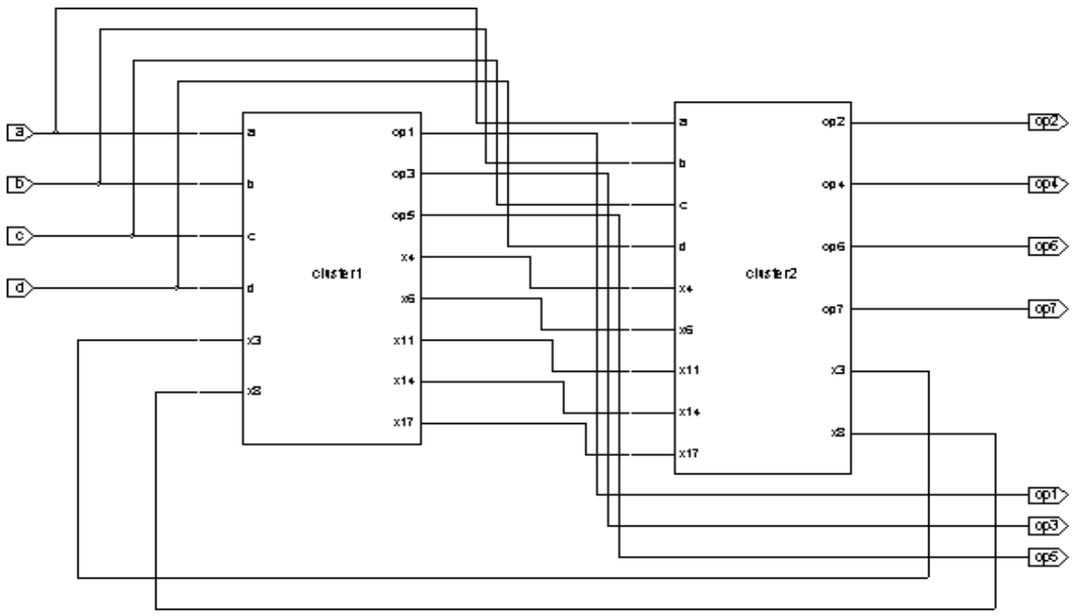



Fig. 22. PAM Partitioned circuit showing 2 sub-circuits

**Results on a Circuit with 15 Nodes**

Example Testing Circuit 2 with 15 nodes:

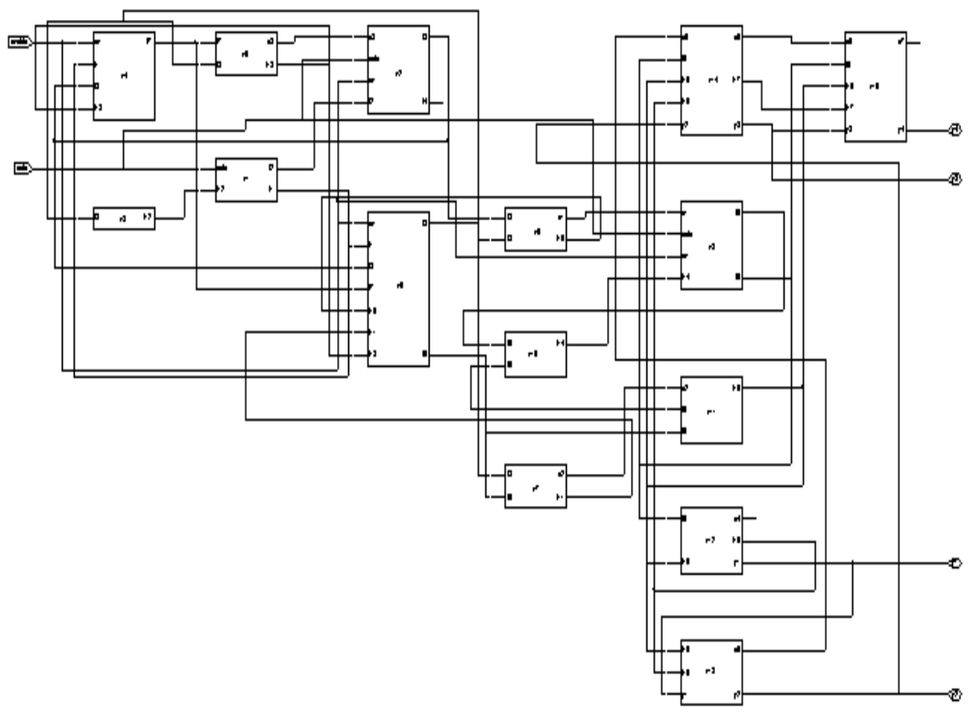


Fig. 23. Circuit before applying partitioning techniques (Rubin, Willy Publications)

IntechOpen

Partitioned circuit obtained after applying Nearest Neighbor Algorithm

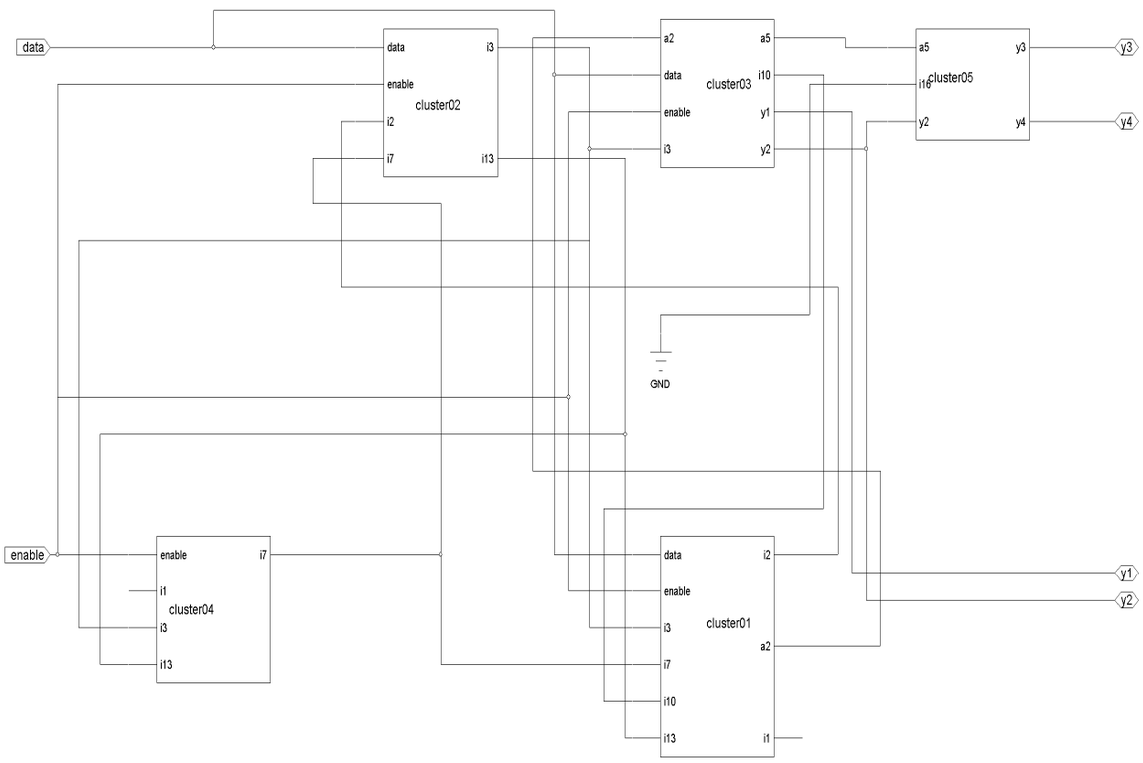



Fig. 24. NNA partitioned circuit showing 5 sub-circuits

Partitioned circuit obtained using Partitioning Around Medoids algorithm

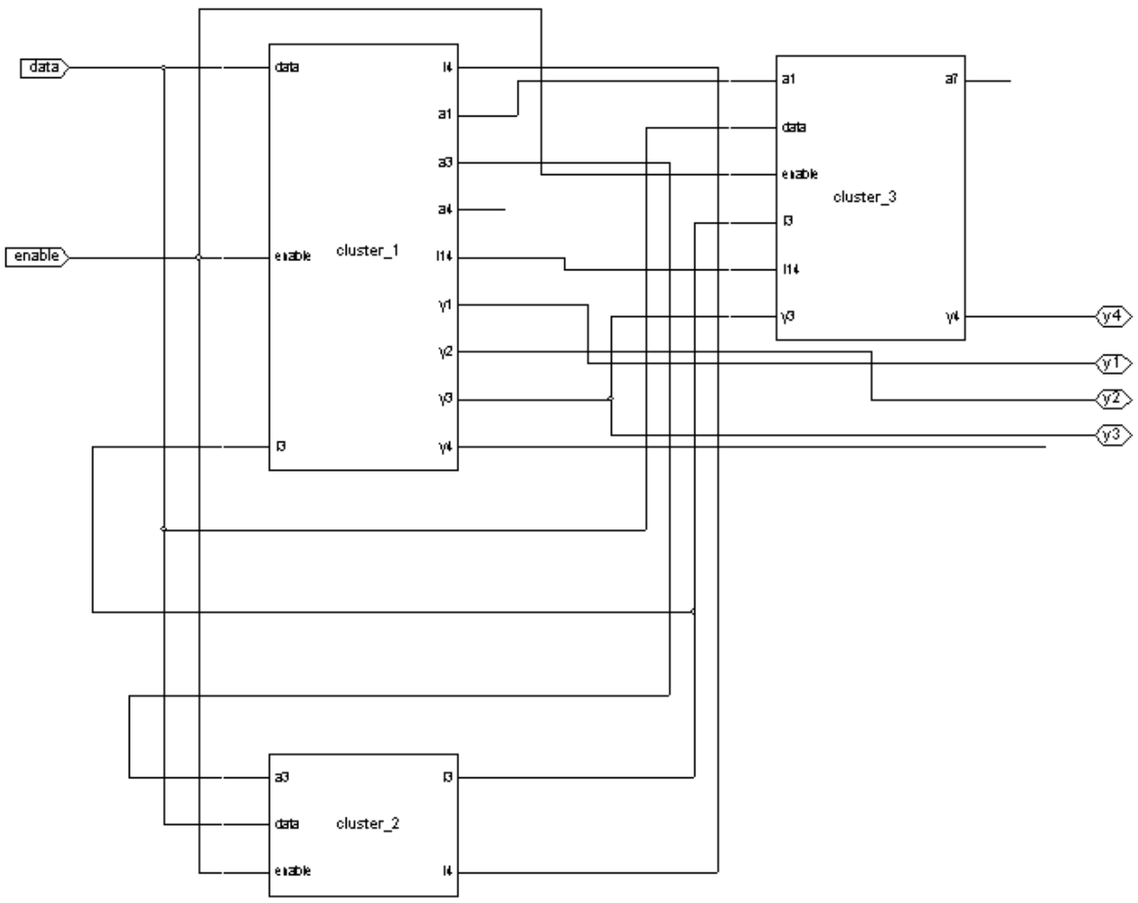



Fig. 25. PAM Partitioned circuit showing 3 sub-circuits

## 8. Conclusion

This section provides observations about the various techniques explained in this chapter with a detailed results based explaination of the Nearest Neighbor and Partitioning Around Medoids Clustering Algorithms.

### 8.1 Memetic approach to circuit partitioning

Memetic algorithm (MA) are population based heuristic search approaches for combinatorial optimization problems based on cultural evolution. They are designed to search in the space of locally optimal solutions instead of searching in the space of all candidate solutions. This is achieved by applying local search after each of the genetic operators. Crossover and mutation operators are applied to randomly chosen individuals for a predefined number of times. To maintain local optimality, the local search procedure is applied to the newly created individuals.

Neuro-Memetic Approach to Circuit Partitioning makes it possible to predict the sub-circuit from circuit with minimum interconnections between them. The system consists of three parts, each dealing with data extraction, learning stage & result stage. In data extraction, a circuit is bipartite and chromosomes are represented for each sub circuit. Extracted sequences are fed to Neuro-memetic model that would recognize sub-circuits with lowest amount of interconnections between them.

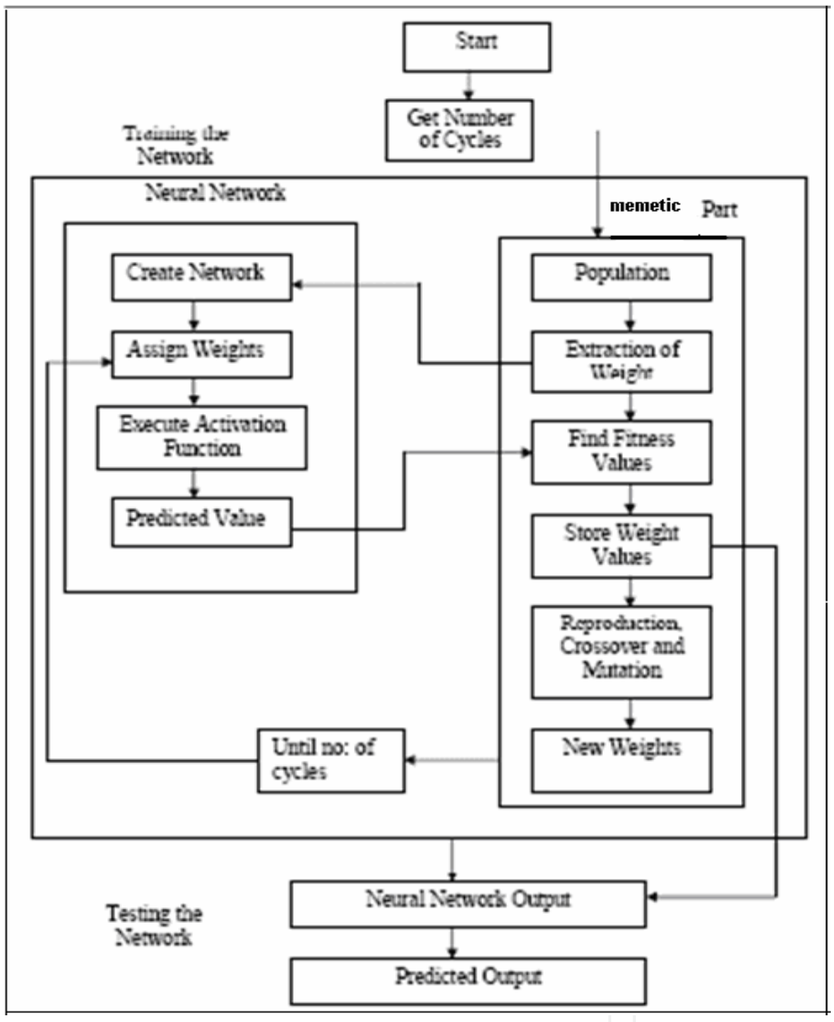



Fig. 26. Working procedure of Neuro-Memetic approach

### 8.2 Neuro-EM model

The system consists of three parts each dealing with data extraction, Learning stage and recognition stage. In data extraction, a circuit is bipartite and partitions it into 10 clusters, a user-defined value, by using K-means (J. B. MacQueen, 1967) and EM methodology (Kaban & Girolami, 2000), respectively. In recognition stage the parameters, centroid and probability are fed into generalized delta rule algorithm separately and train the network to recognize sub-circuits with lowest amount of interconnections between them

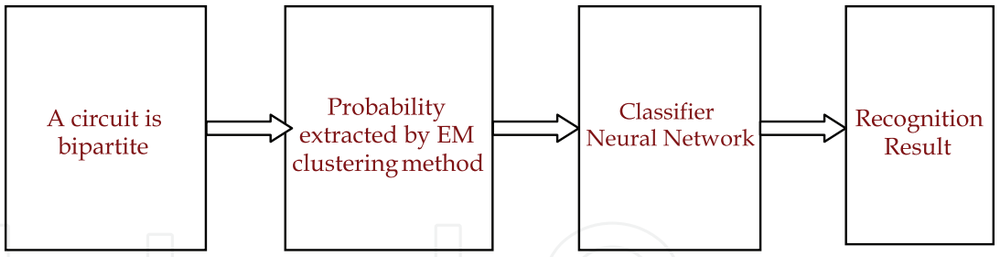



Fig. 27. Block diagram of K-means with neural network

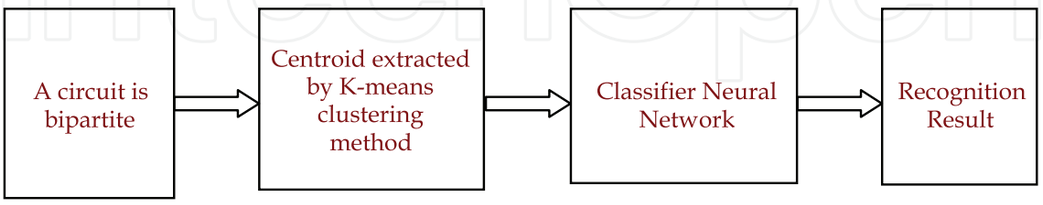



Fig. 28. Block diagram of EM methodology with neural network

In recognition stage the parameters, that is, centroid and probability are fed into generalized delta rule algorithm separately and train the network to recognize sub circuit with minimum interconnection between them

### 8.3 Fuzzy ARTMAP with DBSCAN

A new model for partitioning a circuit is proposed using DBSCAN and fuzzy ARTMAP neural network. The first step is concerned with feature extraction, where it uses DBSCAN algorithm. The second step is classification and is composed of a fuzzy ARTMAP neural network.

### 8.4 Nearest Neighbor and Partitioning Around Medoids clustering Algorithms

Two clustering algorithms Nearest Neighbor (NNA) and Partitoning Around Medoids (PAM) clustering algorithms are considered for dividing the circuits into sub circuits. Clustering is alternatively referred to as unsupervised learning segmentation. The clusters are formed by finding the similarities between data according to characteristics found in the actual data. NNA is a serial algorithm in which the items are iteratively merged into the existing clusters that are closest. PAM represents a cluster by a medoid.

- Criteria Used: Clustering/Unsupervised learning segmentation
- Testing: The algorithms are tested using VHDL,Xilinx xc9500 CPLD/FPGA tool and MATLAB simulator using a test netlist matrix .
- Results and Observations:

As the number of clusters increases, the time taken for PAM increases but is less than Nearest Neighbor algorithm. PAM performs better than Nearest Neighbor algorithm. PAM has been very competent, especially in the case of a large number of cells when compared with Nearest Neighbor. The proposed model based algorithm has achieved sub-circuits with minimum interconnections, for the Circuit Partitioning problem.

| No. of Clusters | Time taken(Secs) | |
|---|---|---|
| | PAM | Nearest Neighbor |
| 2 | 0.045 | 0.5 |
| 4 | 0.067 | 0.7 |
| 6 | 0.075 | 0.8 |
| 8 | 0.087 | 0.9 |

Table 4. Comparison of time taken to partition using PAM and Nearest Neighbor

Completion time: Graphs depict completion time increases proportionately with respect of number of clusters. Nearest neighbor algorithm takes more completion time as number of iterations increase when compared to PAM.

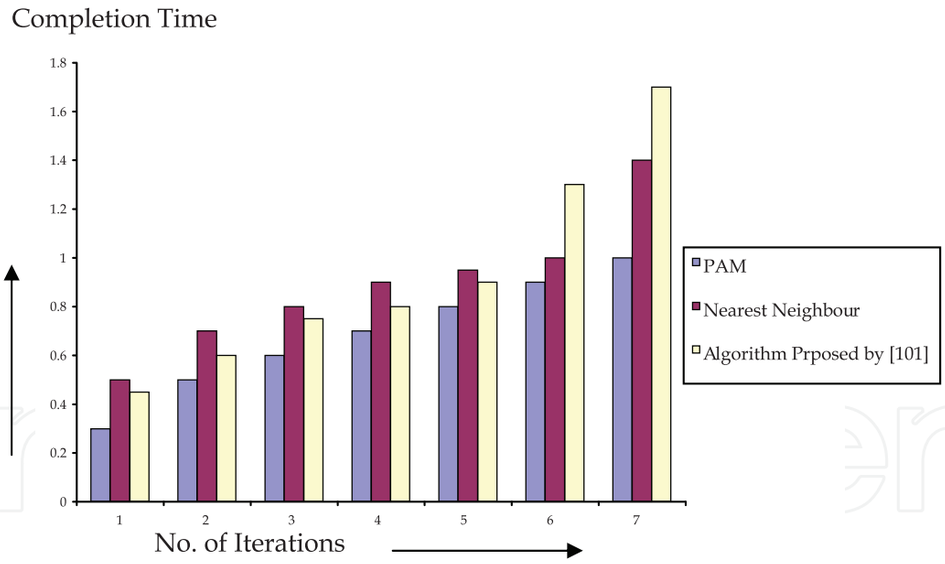



Fig. 29. Graph depicting the Completion time of NNA and PAM compared with algorithm proposed in (Gerez, 1999) over number of clusters

CPU Utilization: A graph depicts CPU utilization increases proportionately with respect of number of iterations. PAM takes less CPU utilization as number of iteration increase compared to NNA algorithm.

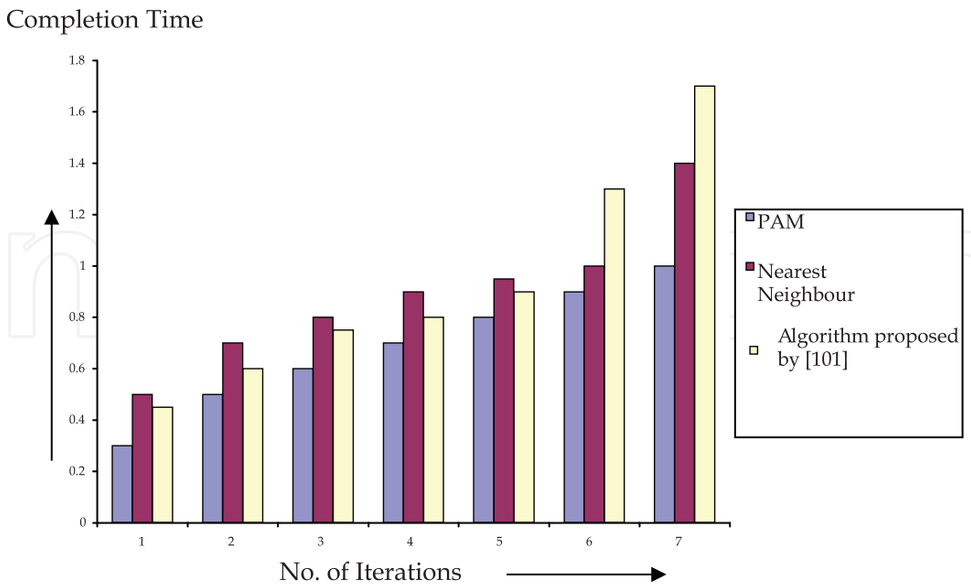



Fig. 30. Graph depicting CPU utilization of NNA, PAM, algorithm proposed in (Gerez, 1999) over number of iterations

From the implementation of the two algorithms, Nearest Neighbor and Partitioning Around Medoids, some fundamental observations are made. There is a reduction of 1 interconnection when a circuit with 8 nodes is partitioned and when a circuit with 15 nodes is partitioned, there is a reduction of 5 interconnections between the sub-circuits obtained using NNA and PAM. Therefore, it is concluded that the number of nodes in a circuit and the number of interconnections are inversely proportional. That is, as the number of nodes in a circuit increases, the number of interconnections between sub-circuits decreases for both partitioning methods. This reduction is not consistent since the complexity of any circuit will not be known a priori. One of the future enhancements would be to analyze the percentage ratio of the number of nodes in a circuit to the number of interconnections that get reduced after the circuit is partitioned.

## 9. Future enhancements

Future enhancements envisaged are using of distance based classification data mining concepts and other data mining concepts, Artificial/ Neural modeled algorithm in getting better optimized partitions.

## 10. Acknowledgement

I would like to acknowledge my profound gratitude to the Management, Rashtreeya Sikshana Samithi Trust, Bangalore.I am indebted to Dr. S.C. Sharma, Vice Chancellor, Tumkur University, Karnataka for his unending support.I would like to thank all my official colleagues at R V College of Engineering and specifically the MCA department staffs for their remarkable co-ordination. Last but not the least, I would like to thank my family members.

## 11. References


J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations", *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1967, pp. 281-297.

Ata Kaban, Mark Girolami, "Initialized and Guided EM-Clustering of Sparse Binary Data with Application to Text Based Documents," ICPR, *15th International Conference on Pattern Recognition (ICPR'00)*, Vol. 2, 2000, pp. 2744.

Shantanu Dutt , Wenyong Deng, "A probability-based approach to VLSI circuit partitioning", *Proceedings of the 33rd annual conference on Design automation*, Las Vegas, Nevada, United States, June 03-07, 1996 , pp.100-105.

N. Krasnogor, J. Smith, "Memetic algorithms: The polynomial local search complexity theory perspective," *Journal of Mathematical Modelling and Algorithms*, Springer Netherlands, ISSN: 1570-1166 (Print) 1572-9214 (online), Vol. 7, Number 1 / March, 2008, pp. 3 - 24.

J. T., "Evolutionary Algorithms, Fitness Landscapes and Search". *Ph.D. thesis, Univesity of New Mexico*, Albuquerque, NM, 1995.

P. Merz and B. Freisleben, "Fitness landscapes, memetic algorithms, and greedy operators for graph bipartitioning," *Journal of Evolutionary Computation*, vol. 8, no. 1, 2000, pp. 61–91.

T. Jones, "Crossover, macromutation, and population based search," *in Proceedings of the Sixth International Conference on Genetic Algotihms*, M. Kauffman, Ed., 1995, pp. 73–80.

N. Krasnogor, P. M. L´opez, E. de la Canal, D. Pelta, "Simple models of protein folding and a memetic crossover," *in Exposed at INFORMS CSTS, Computer Science and Operations Research: Recent Advances in the Interface meeting*, 1998.

Christopher M. Bishop, "Neural Networks for Pattern Recognition", *Oxford Univ Press*, ISBN: 0198538642,1995.

N. Krasnogor, P. Mocciola, D. Pelta, G. Ruiz, W. Russo, "Arunnable functional memetic algorithm framework," in *Proceedings of the Argentinian Congress on Computer Sciences*. Universidad Nacionaldel Comahue, 1998, pp. 525–536.

D. Holstein, P. Moscato, "Memetic algorithms using guided local search: A case study," *in New Ideas in Optimization*, D. Corne, F. Glover, M. Dorigo, Eds. McGraw-Hill, 1999.

Arthur Dempster, N Laird, D B Rubin, "Maximum likelihood estimation for incomplete data via the EM algorithm", Journal of the *Royal Statistical Society* (B) 39, 1977, pp.1-38.

Witten I, Frank E, "Data Mining: Practical Machine Learning Tools and Techniques", *Second Edition, Morgan Kaufmann*, San Francisco, 2005.

Dimitris Karlis, "An EM algorithm for multivariate Poisson distribution and related models", *Journal of Applied Statistics*, Vol. 30, No. 1, 2003 , pp. 63-77(15)

Carpenter, G.A, "Distributed learning, recognition, and prediction by ART and ARTMAP neural networks", *Neural Networks*, 10:, 1997, pp. 1473- 494.

Sung Ho Kim, "Calibrated initials for an EM applied to recursive models of categorical variables", *Computational Statistics & Data Analysis*, Vol. 40, Issue 1, July 2002, pp. 97-110 .

Gyllenberg M, Koski T, Lund T, "Applying the EM-algorithm to classification of bacteria", *Proceedings of the International ICSC Congress on Intelligent Systems and Applications*, 2000, pp. 65-71.

Carpenter, G.A., Grossberg, S., "Adaptive Resonance Theory, In Michael A. Arbib" (Ed.), *The Handbook of Brain Theory and Neural Networks*, Second Edition,Cambridge, MA: MIT Press, 2003, pp. 87 – 90.

Grossberg, S., "Competitive learning: From interactive activation to adaptive resonance", Cognitive Science

Carpenter, G.A, Grossberg, S., "ART 2: Self-organization of stable category recognition codes for analog input patterns", *Applied Optics, 26(23)*, 1987, pp. 4919-4930.

Carpenter, G.A., Grossberg, S., Reynolds J.H., "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network", *Neural Networks (Publication),* 4, 1991, pp. 565-588.

Carpenter, G.A., Grossberg, S., Rosen, D.B., "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", *Neural Networks (Publication),* 4, 1991, pp. 759-771.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen D.B., "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps", *IEEE Transactions on Neural Networks*, 3, 1992, pp. 698-713.

A. Canuto, G. Howells, M. Fairhurst, "An investigation of the effects of variable vigilance within the RePART neuro-fuzzy network," Journal of Intelligent and Robotic Systems, 29:4, 2000, pp. 317-334.

A. Dubrawski, "Stochastic validation for automated tuning of neural network's hyper-parameters," *Robotics and Automated Systems*, 21, 1997, pp. 83-93.

P. Gamba and F. DellAcqua, "Increased accuracy multiband urban classification using a neuro-fuzzy classifier," *International Journal of Remote Sensing*, 24:4, 2003, pp. 827-834.

C. P. Lim, H. H. Toh, T. S. Lee, "An evaluation of the fuzzy ARTMAP neural network using offline and on-line strategies," *Neural Network World, 4*, 1999, pp. 327-339.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., and Rosen, D.B., "Fuzzy ARTMAP: neural network architecture for Incremental supervised learning of analog multidimensional maps", *IEEE Transactions on Neural Networks*, 1992, 3:, pp. 698-713.

Caudell, T.P., Smith, S.D.G., Escobedo, R., Anderson, M, "NIRS: Large scale ART–1 neural architectures for engineering design Retrieval", *Neural Networks, 7:*, 1994, pp. 1339-1350.

Benno Stein, Michael Busch, "Density based cluster Algorithms in Low-dimensional and High-dimensional Applications", *Second International Workshop on Text- Based information Retrieval (TIR 05)*, Stein, Meyer zu Eißen (Eds.) Fachbericht,e, Informatik, University of Koblenz-Landau, Germany, ISSN 1860-4471c, pp. 45-56.

Busch. Analyse dichtebasierter Clusteralgorithmen am Beispiel von DBSCAN und MajorClust. Study work, Paderborn University, Institute for Computer Science, March 2005.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise". *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*: AAAI Press, 1996, pp. 226-231.

Tan, S.C., Rao, M.V.C., Lim C.P, "An adaptive fuzzy min-max conflict-resolving classifier," *in Proceedings of the 9th Online World Conference on Soft Computing in Industrial Applications, WSC9*, 20 September – 8 October 2004.

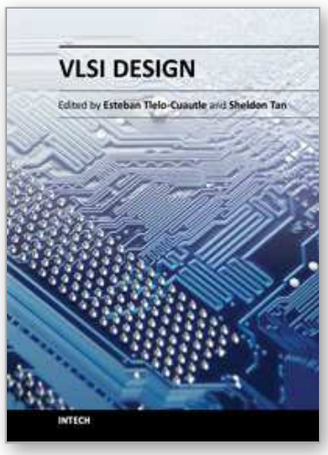
Dagher. I, Georgiopoulos. M, Heileman. G.L., Bebis, G, "An ordering algorithm for pattern presentation in fuzzy ARTMAP that tends to improve generalization performance," *IEEE Trans Neural Networks*, vol. 10, 1999, pp. 768-778.

Jianhua Li, Laleh Behjat, "A Connectivity Based Clustering Algorithm with Applications to VLSI Circuit Partitioning", *IEEE Transactions on Circuits and Systems-II.Express Briefs*, Vol.53, No.5, May 2006.

Steven M Rubin, "Computer Aids for VLSI Design", Second Edition, Willy Publications.

Sabin H Gerez, "Algorithms for VLSI design automation", University of Tacut, department of electrical engineering, Netherlands, John Wiley & Sons, 1999, pp. 1-15.

IntechOpen




**VLSI Design**

Edited by Dr. Esteban Tlelo-Cuautle

ISBN 978-953-307-884-7
Hard cover, 290 pages
**Publisher** InTech
**Published online** 20, January, 2012
**Published in print edition** January, 2012

This book provides some recent advances in design nanometer VLSI chips. The selected topics try to present some open problems and challenges with important topics ranging from design tools, new post-silicon devices, GPU-based parallel computing, emerging 3D integration, and antenna design. The book consists of two parts, with chapters such as: VLSI design for multi-sensor smart systems on a chip, Three-dimensional integrated circuits design for thousand-core processors, Parallel symbolic analysis of large analog circuits on GPU platforms, Algorithms for CAD tools VLSI design, A multilevel memetic algorithm for large SAT-encoded problems, etc.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

K.A. Sumithra Devi (2012). Algorithms for CAD Tools VLSI Design, VLSI Design, Dr. Esteban Tlelo-Cuautle (Ed.), ISBN: 978-953-307-884-7, InTech, Available from: http://www.intechopen.com/books/vlsi-design/algorithms-for-vlsi-cad-tools

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen