Classification Algorithms in Achieving Partitioning Optimization for VLSI Applications

Dr .Sumithra devi K.A. and Vijayalakshmi.M.N

***Abstract*—The relevance of VLSI in performance computing, telecommunications, and consumer electronics has been expanding progressively, and at a very hasty pace. In order to build complex digital logic circuits it is often essential to sub-divide multi million transistors design into manageable pieces.Circuit partitioning is a general approach used to solve problems that are too large and complex to be handled at once. In partitioning, the problem is divided into small and manageable parts recursively, until the required complexity level is reached. In the area of VLSI, circuit complexity is rapidly multiplying together with the reducing chip sizes, the integrated chips being produced today are highly sophisticated. There are many diverse problems that occur during the development phase of an IC that can be solved by using circuit partitioning which aims at obtaining the sub circuits with minimum interconnections between them. This paper aims at circuit partitioning of VLSI application using classification techniques, that is, Decision Tree Algorithm and K-Nearest Neighbors Algorithm. These two algorithms were tested for partitioning optimization on a 3-bit Priority Encoder and a 4x2 SRAM sample circuits and implemented using VHDL. The tested results shows that the K-Nearest Neighbor algorithm yields better subcircuits than the Decision Tree Algorithm.**

***Index Terms*—Circuit Partitioning, Decision Tree Classification Algorithms, K-Nearest Neighbor algorithm,**

I. Introduction

Advances in semiconductor technology and in the integration level of integrated circuits have enhanced many features, increased the performance; improved reliability of electronic equipment, and at the same time reduced the cost, power consumption and system size. As size and complexity of digital system has increased, more computer aided design tools are introduced into hardware design processes. The designers extensively rely on software tools for nearly every aspect of the development cycle, central to VLSI design automation, and one that has attracted a great deal of interest; a recent survey [1] lists almost 200 papers on the subject. In order to reduce the complexity of the design process, several intermediate levels of abstraction are introduced. Partitioning a circuit is necessary if it is too large to be accommodated on a single chip.A number of heuristic algorithms have been developed over the past many decades for various physical design problems. Different heuristic techniques, which are generally employed for solving different physical design sub-problems, are Iterative improvement algorithms, such as

- Kernighan-Lin (KL) algorithm [3]
- Fiducccia-Mattheyses (FM) algorithm [4]
- Simulated Annealing
- Genetic Algorithms

II. Circuit Partitioning

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 [2].This paper proposes a computing approach to unravel circuit partitioning problem. It is trained to learn useful sub-circuits with lowest amount of interconnections between them. In this paper an attempt is made to achieve minimum interconnections is obtained by applications of two classification algorithms K-Nearest neighbor and Decision Tree. These two classification algorithms are applied to 3-bit Priority Encoder and a 4x2 SRAM sample circuits. Results show that K-Nearest Neighbor gives better subcircuit clusters with minimum interconnections. Both the classification algorithms are run using VHDL tool for testing.

III. Implementation

The implementation includes three stages consisting of data extraction, partitioning and result. In data extraction, a sample circuit of 3-bit Priority Encoder and a 4x2 SRAM circuit are considered for partitioning concept of the circuit. The 3-bit Priority Encoder has 8 inputs and 3 outputs and 4x2 SRAM has 5 inputs and 2 outputs. The circuits considered are shown below in Fig 1 and Fig 2.

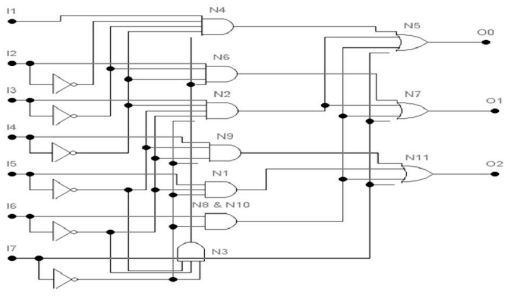

Figure 1. Priority Encoder with Node numbers

Dr .Sumithra devi K.A. *Member, IACSIT No.80334046*,
Vijayalakshmi.M.N Member, *IACSIT* 80332289

IV. Overview of Classification Algorithms Used

Classification is referred as supervised learning segmentation. The classes are formed by obtaining the similarities between data according to characteristics found in the data. The input data for a classification task is a collection of records. Each record, which is also known as an instance or an example, is characterized by a tuple (x, y), where x is the attribute set and y is a special attribute designated as the class label. Classification is also defined as the task of learning a target function f that maps each attribute set x to one of the predefined class labels y. The target function is also known as a classification model[7]. A classification technique is a systematic approach to building classification models from a an input data set. Each technique employees a learning algorithm to identify a model that bests fits the relationship between the attribute set and class label of the input data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before. Therefore a key objective of the learning algorithm is to build models with good generalization capacability i.e, models that accurately predict the class labels fo previously unknown records. In this paper we are using K-Nearest neighbor and Decision Tree classification algorithms for implementing circuit partitioning.

A. K-Nearest Neighbor Algorithm s

K-Nearest Neighbor (KNN) classification methods can be considered as one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. K- Neighbor classification was developed from the need to perform discriminate analysis when reliable parametric estimates of probability densities are
unknown or difficult to determine. The k-nearest-neighbor classifier is commonly based on the Euclidean distance
between a test sample and the specified training samples. The algorithm described is as follows

Algorithm

Input :

T- Training Data
K- Number of Neighbors
t- Input Tuple to Classify

Output :

C- Class to which t is assigned

```
KNN //Algorithm
N=0; // Find set of neighbors, N, for t.
for each d T do
    if |N|<= K, then
        N=N U (d);
    else
        if u N such that sim(t,u)<=sim(t,d), then
            N = N (d);
            N = N U (d);
End
C=class to which the most u N are classified
```

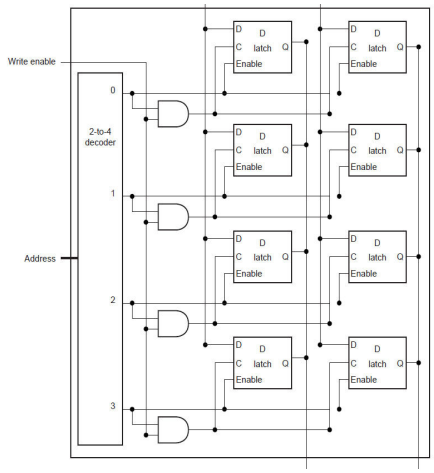

Figure 2 Detailed SRAM Circuit

B. Decision Tree Algorithm

The decision tree hs three types oif nodes: A Root node that has no incoming edges and zero or more out going edges. A Internal nodes, each of which has exactly one incoming edge and two or more outgoing edges. Leaf or terminal nodes, 3each of which has exactly one incoming edge and no outgoing edges[7]. Each leaf node in a decision tree is assigned a class label. The non terminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics. Classifying a test record is straightforward once a decision tree has been constructed. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test. This will lead us either to another internal node, for which a new test condition is applied, or to a leaf node. The class label associated with the leaf node is then assigned to the record. The algorithm is described below

Algorithm:

```
if stopping_cond(E, F) = true then
        leaf = createNode().
        leaf.label = Classify(E).
        return leaf.
else
        root = createNode().
        root.test_cond = find_best_split(E, F)
let V = { v | v is possible outcome of root.test_cond }.
for each v V do
   Ev = { e | root.test_cond(e) = v and e E }
   child = TreeGrowth(Ev, F)
  add child as descendent of root and label the
  edge (root à child) as v.
end for
end if
return root.
```

The input to this algorithm consists of the training records E and the attribute set F. The algorithm works by recursively selecting the best attribute to split the data and expanding the leaf nodes of the tree until the stopping criterion is met

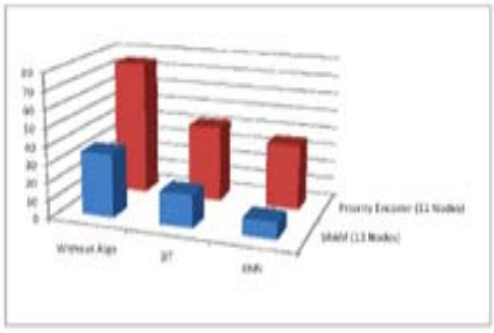

Figure 3 Graph Depicting Comparison of two algorithms

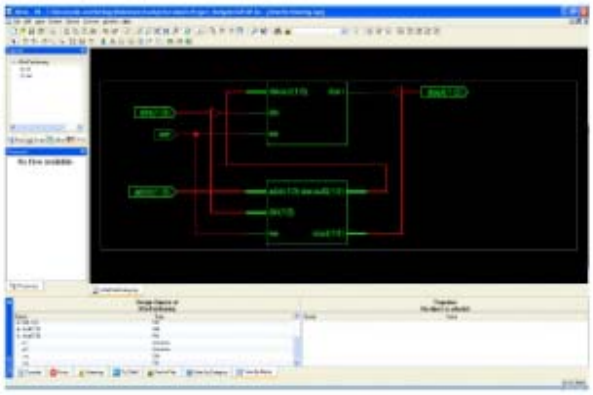

Figure 4 Screen Shot depicting the portioned circuit after applying KNN algorithm for Priority Encoder

V. Results

In data extraction, the circuit which is considered is represented by an adjacency matrix where the values represent the distances between the nodes. The adjacency matrix is fed into algorithm which gives the sub-circuits with interconnections between them. The partition is tested on the circuit. Table 1 and Table 2 gives the performance of Decision tree and K-Nearest Neighbour algorithms for the sample circuits, which is tested on Xilinx 6.1 Navigator tool using VHDL, screenshots of the results for both the sample circuits are as shown in the figure 4 and figure 5.

VI. Conclusions

In this paper the Circuit Partitioning which plays a critical role in physical design of an integrated circuit with minimum interconnections is implemented using two classification algorithm Decision Tree and K-Nearest Neighbour. Experimental results show that K-Nearest Neighbour gives minimum interconnections compared to Decision Tree for the same input. The number of interconnections in the circuit is 18 in the case of DT and 9 in the case of KNN. After analyzing the results, we find that KNN method performs better to satisfy constraints, and achieve the objective of minimum circuit interconnections when compared to DT algorithm. The graph below, Fig 3, shows the comparison of DT and KNN algorithms.

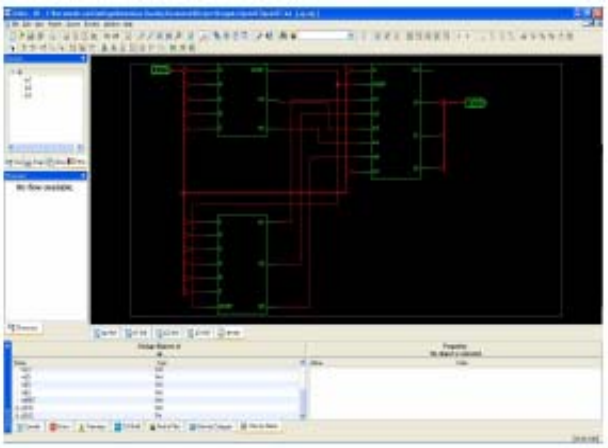

Figure 5 Screen Shot depicting the portioned circuit after applying KNN algorithm for 4X 2 SRAM

TABLE I Priority Encoder Results

Initial No of Interconnects	75
After Distance Based Classification	42
Final Optimization Using K Nearest Neighbor Approach	36

TABLE 2 SRAM Results

Initial No of Interconnects	35
After Distance Based Classification	18
Final Optimization Using K Nearest Neighbor Approach	9

References

[1] C.J. Alpert and A.B. Kahng, "Recent Directions in Net list Partitioning: A Survey," Integration: the VLSI Journal, 19(1-2), 1995, pp. 1 - 81.

[2] Pinaki Mazumber, Elizabeth M.Rudnick,"Genetic Algorithms for VLSI Design, Layout &Test Automation", Prentice Hall PTR, 1998

[3] B.W.Kernighan and S.Lin,"An Efficient Heuristic Procedure for partitioning Graphs", Bell System Tech Journal, vol 49, Feb.1970, pp.291-307

[4] C.M.Fiduccia and R.M. Mattheyses,"A Linear-time heuristic for improving Network partitions", Proc. ACM/IEEE Design Automation Conf., 1982, pp 175- 181

[5] Jan Madsen, Jesper Grode, and Peter V. Knudsen. Hardware/Software Partitioning using the LYCOS System, chapter 9. Hardware/Software Codesign: principles and Practice. Kluwer Academic Publishers, Netherlands, 1997.

[6] Martin Ester,Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)

[7] Pang-Ning Tan, Michael Steinbach, Vipin Kumar,"Introduction to Data Mining",2nd Edition, Pearson Publications