



DETERMINISTIC AND PROBABILISTIC MODELS ON VLSI CELL PLACEMENT-A SURVEY

¹R.MANIKANDAN, ²M.THIYAGARAJAN, ³P.SWAMINATHAN

¹Senior Assistant Professor, School of Computing, SASTRA University, Thanjavur, India -613401

²Visiting Professor, Department of Mathematics, AMRITA University, Coimbatore, India -641112

³Dean, School of Computing, SASTRA University, Thanjavur, India -613401

E-mail: manikandan75@core.sastra.edu, srmanimt75@yahoo.com, deanpsw@sastra.edu

ABSTRACT

General VLSI Cell placement has gone through different versions depending upon the particular applications. The area under modern challenges of VLSI design throw light on Power minimization, Thermal capacity and Area occupation. Thus Utility function, Renewal reward and Hypergraph setup are utilized in our discussion. A brief review is given in this paper

Keywords: VLSI Cell Placement, Renewal Reward Process, Utility Function, Hypergraph.

1. INTRODUCTION

Major VLSI Physical design issues are addressed at different points with different perspective. Graph theoretical algorithms, deterministic and non deterministic optimization problems and probabilistic models are studied in this area to give complete solution to problem arising in this direction. We have analysed the problem through hyper-graph, utility function, renewal reward theory and few optimization techniques. The paper is presented in three sections. In section I, we deal with relevance of utility function and generalization in VLSI Cell placement. In section II, We deal with graph theoretical algorithms for constraint placement. In section III, we deal with hypergraph for VLSI Cell partitioning.

2. SECTION -I

VLSI cell placement problem can be regarded as an information processing through reinforced learning rule. Since the movements are governed by non-linear dynamics and chaos the explorations at each stage is a dynamic programming process. During this learning process we do not have the knowledge of the previous history and assuming that the future depends on the present state, we get the model to satisfy Markov-property. Besides this we have acquired a partial breakthrough leading to a reward. This reward is to be regarded as a function of time. Transit from different states of the system is governed by a certain weight attached learning rule. We are left with pricing our options and take care of reward exploration leading to

unlimited transits as the circuitry in VLSI Design. Here we coin the problem as non-linear dynamics and chaos information processing in neural network [1,2]

We can divide the VLSI circuit design problem into sub problems and solve each sub problem independently to give more effective and efficient technique to handle existing difficulties and design complexities. We address the problem of Channel Routing as information processing neural network through non linear dynamics and uncertainly.

Temporal Difference (TD) Method

TD methods are based upon the following simple recursive relation: When we are trying to estimate Q experimentally, the estimates for different states are related as

$$Q(s_t, a_t) = \sum_{k=0}^{\infty} \gamma^k r_{k+t+1} = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}), \text{ and}$$

$$\text{therefore } \Delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) = 0.$$

If $\Delta_t \neq 0$, it means that the present estimates have to be modified. So, the methods look as follows. One uses time-dependent estimates $Q(s, a, t)$ of action values. The initial guesses $Q(s, a, 0)$ can be chosen arbitrarily. Then one chooses a policy π , and after every step modifies the action values iteratively:

$$Q(s, a, t+1) = Q(s, a, t) + a_t e(s, a, t) \Delta_t.$$

This relation contains two new factors – the learning rate a_t and the so-called eligibility traces

$e(s, a, t)$, which we shall describe a little later. For the evaluation of the correction of the correction term Δ , there are two major methods.

a) "Sarsa" (the name reflects the sequence $s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}$ used for Δ evaluation) [3]

$$\Delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}, t) - Q(s_t, a_t, t).$$

It provides the estimates of Q for the current policy. Nonetheless, if one uses the current values of Q for the policy formation, e.g., at the moment t chooses the action a with the largest current $Q(s_t, a, t)$, then it proves that the method admits policy improvement as well.

(b) "Q-learning",

$$\Delta_t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a, t) - Q(s_t, a_t, t).$$

This method converges to the action values Q^* for the optimal policy π^* almost independently of the current policy π . Knowing Q^* it is easy to obtain the optimal policy π^* itself.

Let us consider eligibility traces which are reflexes related to events neural and assumptions that are separated by a set of time intervals. Neural events correspond change in weight after neural reputation the essential events occur which is proportional to signal, for some time for resetting the corresponding connection can change until the new settings can take place.

It is necessary in that the information about the environment is complete the action value Q , the optimal policy can be calculated. This is to use activities with the largest Q . Even if some of the Q estimated accurately. There always remind the possibilities the best option might not be tried. This policy is called greedy policy. Among the best known action with probability $1-e$ other action with probability $e/1-n$. We conclude neural networks with reinforced learning can be used to study non linear dynamics with information processing.

Renewal reward process

In the context of channel routing, the decisions involve assigning nets to tracks such that no horizontal and vertical constraints are violated and the number of tracks is minimum. In the previously developed search techniques based routers, the neighbourhood solutions are generated by randomly selecting nets and tracks. However, random moves do not guarantee convergence in a reasonable time and the algorithm is highly likely to get trapped in a local minima. In this work, a variety of problem

domain information are combined using Markov process that, nets and tracks are selected based on their respective rewarding. The net reward renewal function express information about the goodness for assigning a net to a track. The track reward combines information about the goodness of cluster of nets assigned to a track and about the sparsity of the track.

We now present how the channel routing problem-domain information can be mapped into Markov process and renewal reward process. [4]

Let $\{X_n\}$ denote the stochastic poisson-process with parameter λ to represent the tip of channel routing and process $\{Y_n\}$ is a poisson process with parameter μ to denote the cell design. The cumulative process $Y(t) = \sum_1^{w(t)} Y_n$ is the reward process. Then we can obtain the complete description of channel route and cell placed VLSI design is got as a limit of t tends to infinity.

A set of benchmark problems taken from an existing paper [5] and the difficult problem of Deutsch are used to evaluate the performance of the algorithm. The optimal routing width of each benchmark is known. The statistics for these benchmarks are shown in table 1. Several scenarios were conducted using these benchmarks. In each scenario, the performance of the algorithm was evaluated based on one of the following criteria (1) convergence behavior; (2) effectiveness of incorporating Tabu Search; (3) effectiveness of using utility functions to select the best moves. Due to the fact that the SE algorithm is stochastic in nature and to ensure the validity of the approach, the algorithm was executed for 20 trials in each scenario.

The first scenario was designed to investigate the convergence of the algorithm.

Table 1: Statistics for the different benchmarks

Benchmark	No. of nets	Global optimum
<i>ex1</i>	21	12
<i>ex3a</i>	45	15
<i>ex3b</i>	47	17
<i>ex3c</i>	54	18
<i>Deutsch ex.</i>	72	28

The routing of the difficult Deutsch problem is illustrated in figure (1). Furthermore, as shown in figure (2), the valley representing the optimal routing width was explored at an average number of generation of 172. The algorithm was executed with the SETS (SE and TS) hybrid as the search engine. Also, utility functions are used to select best moves during exploration of the solution space. Figure (2) illustrates the variation of the average cost versus the generation number for the Deutsch difficult example. By observing the average cost contour, one can easily see that the algorithm climbs hills and descends valleys of the solution space. This suggests that the algorithm explores the solution space effectively and convergence to the global optimum is very likely.

The second scenario was designed to ascertain the effectiveness of combining SE and TS as a hybrid search engine (SETS hybrid). In this scenario, two experiments were conducted. In the first experiment, the algorithm employed only Se as a search engine, and in the second experiment the SETS hybrid was employed as a search engine. Moves were randomly selected (i.e., nets and destination tracks are randomly chosen). The algorithm converged to the global optimal routing width in both experiments, except for the extra benchmark for which the optimal answer obtained in the first experiment, is one track beyond the global optimum. For all the benchmarks, the number of generation, AVG-GEN, needed to converge to the optimal solution and, OPT-SOL, in the second experiment are less compared to the first experiment (see Table 2). This observation demonstrates the significance of SETS hybrid as a search engine.

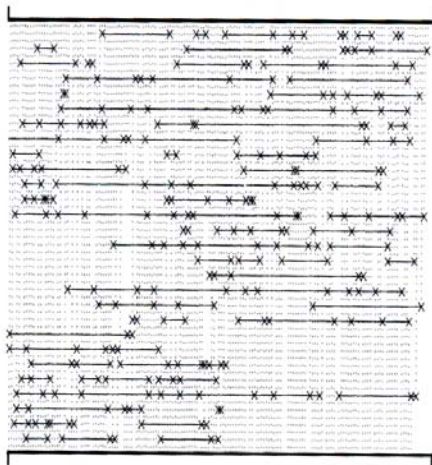


Figure 1. The routing of Deutsch's difficult example

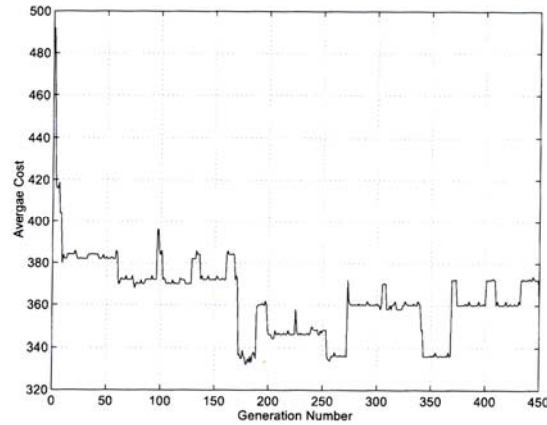


Figure 2. The variation of the average cost value as the algorithm progresses over generations for the difficult Deutsch problem

Table 2: For all the benchmarks, average number of generations (AVG-GEN) required to converge to an optimal solution (OPT-SOL) when the search engine is (i) only SE; (ii) SETS hybrid and utility functions (SETS-UTFN).

Benchmark	AVG-GEN			OPT-SOL		
ex1	4	4	4	12	12	12
ex3a	335	160	310	16	16	15
ex3b	297	227	206	17	17	17
ex3c	67	56	45	18	18	18
Deutsch ex.	314	217	172	28	28	28

In the third scenario, the effectiveness of using utility functions in selecting the best moves is investigated. The algorithm is executed with utility functions used to choose candidate nets and candidate tracks for a move. The results obtained are again reported in table 2. The global optimum for each benchmark is obtained. Also, the AVG-GEN required to obtain the global optimum is smaller compared to the previous scenarios. This indicates that using of utility functions to determine best moves (rather than relying on random moves) in generating neighborhood solution is a quite effective approach. The major limitation of the SETS channel router is in the choice of the control parameters; i.e., Tabu Length, stopping parameter, control parameter and the threshold criteria.

Regarding the Aspiration criterion used, again, our experience shows that such a criterion is quite effective in forcing the SE to backtrack previous solutions and refine those search regions.

3. SECTION-II

In this section, we firstly describe the significant of graph theoretical algorithms. In understanding the details of cell placement process in VLSI, we come to the conclusion that this problem is an NP-Complete problem. It is practiced to search for relevant NP-Complete problems which can be put into one-to-one correspond of similar NP-Complete problem. One of the graph theoretical algorithms which can be identified to correlate our VLSI design and placement is the vertex cover problem.

The Constrained placement problems deal with the computation of an optimal arrangement of items on a planar site. The objective function for these optimization problems is based on the overall rectangular area of occupied space and on addition of all terms that reflect problem-specific constraints. The basic variants of these problems are unconstrained two-dimensional packing problem and the quadratic assignment problem. In the case of the packing problem, a set of rectangular blocks has to be arranged such that no blocks overlap each other. The area of the rectangle circumscribing all blocks has to be minimal, hence the optimal packing pattern is that with minimal waste inside the enveloping rectangle. In the quadratic assignment problem [6], a set of items has to be matched to fixed arranged bins. A flow matrix defines the connectivity between the items. The objective is to find a mapping with minimal flow costs, these being the sum of the products of flow and distance between each pair of items. The quadratic assignment problem is an NP-hard optimization problem [7].

To deal with different often conflicting objectives divide and conquer approach is typically used. Multistage approaches compute optimum arrangement of blocks are based on the connectivities. Connectivity between the blocks considered as structured graph theoretical problem.

Threshold Device Networks For Generalized Vertex Cover Problems

The primary theoretical contributions of this paper are expressed in three theorems, each of which contributes to be eventual development of

methodologies for effective on-chip spare row/column allocation in VLSI arrays. Here we identify and analyze a class of threshold device networks which is proven applicable to the GVC problem.

Consider a network of threshold devices whose energy function is of the form

$$E_{VC}(S) = -A \sum_{i=1}^N \sum_{j=1}^N S_j C_{ij} S_j$$

where each C_{ij} is determined by the connection matrix of some undirected graph G with no self-loops, and where A is any positive constant. If a correspondence is established between device I of the network and vertex i of G for $1 \leq i \leq N$, and if a subject V_0 is defined of which vertex i is considered a member whenever $S_i = 0$, then we have the following lemma.

Lemma: 1

Let H be a network of laterally connected threshold devices whose energy function is given by E_{VC} . A necessary and sufficient condition for any state to be a local minimum of E_{VC} and hence a stable state of H , is that its associated vertex set V_0 be a vertex cover of G .

Lemma: 2

Let H be a network of laterally connected threshold devices whose energy function is given by E_{MVC} and whose parameters A and B_i are chosen so as to obey the inequalities

$$A > \frac{B_i(2|L_i|-1)}{2} \quad \text{and} \quad B_i > 0 \quad \text{for}$$

$1 \leq i \leq K$.

A necessary condition for any state of H to be stable is that its associated V_0 be a vertex cover of G .

Let H be a network of laterally connected threshold devices which fulfills the

requirements of Lemma 2, and let U denote the union of all sets L_i of threshold devices for which $B_i > 0$. A necessary and sufficient condition for any state of H to be stable is that its associated vertex set V_0 be a vertex cove of G which is minimal over the set of vertices corresponding to U .

Iterative Clustering Based On Connectivity

While slicing trees and shape functions deal with the packing aspect of the optimization problems, another heuristic is necessary to take into account those constraints based on connectivity.

When constructing a solution for these problems, care must be taken that highly connected blocks are placed near each other on the layout site. During the composition of a slicing layout, blocks (or meta-blocks) are iteratively paired. This corresponds to composing an inner node of the slicing tree by joining two leaves (or subtrees).

At the beginning of the construction, globally good pairings are identified in the set of all blocks. These build the lowest level of the slicing tree. A good technique for successively pairing items according to a quality function is the iterated *matching* heuristic, which was introduced by Fritsch and Vornberger [8] It is based on the graph algorithmic computation of a *maximum weight matching* on a complete graph. The vertices of this graph represent the items to be paired, and each edge is weighted with the value of the quality function for the corresponding pairing. A matching in this graph is a set of node-disjunct edges, and the weight of a matching is the sum of the weights of all edges in this set.

In the case of the facility layout problem, an edge is weighted according to the flow between the facilities represented by the adjacent vertices. For the VLSI layout generation problem, the quality function is based on the number of terminals that have to be connected by signal nets between both cells. A maximum weight matching corresponds to a set of optimal pairings such that a globally maximal number of terminals can be connected inside the resulting meta-blocks. Since paired blocks are adjacent on the final layout, the maximum weight matching ensures short wiring lengths in the case of macro-cell layout generation and low partial flow-cost terms for the facility layout problem.

We have used the graph-theoretical algorithm namely, generalized vertex cover problem to describe the VLSI cell placement problem process. This procedure has been illustrated through two specific applications one on one-line chip embedded system and another with mobile secure communications.[9]

4. SECTION-III

The Main problem of partitioning in VLSI Design and other related domain of study can be posed as getting sub domain of equal size of the vertices in a given hypergraph. A Variety of knowledge based

algorithms have been developed which give different cost-quality tradeoffs. The edges at consecutive cells dominate each other to have incident one, two or more creating cluster like situation which can be modeled as a hypergraph. The contribution of links to specific node is relative important and thus demands for probabilistic algorithms to develop on stochastic function. Iterative refinement algorithms have been developed by Kernighan-Lin (KL) and Fiduccia-Mattheyses [10,11,12]

Earlier Researchers have placed the problem in searching for Max-Min cut on graphical representation of the design. Here, We give hyper graph concepts explained through discrete time and space sequence of random variables forming Markov chain [13]. The classification of nodes studied through an analysis on variety of classified states of the Markov chain fitted to the situation.

Problem Specification

Formally, the k -way min-cut partitioning problem can be stated as follows. Let a circuit C be represented by a hypergraph or netlist $G = (V, E)$, where V is the set of nodes that represent the nets of the circuit. Each hyperedge or net connects two or more nodes together; generally the output of a node is connected to the inputs of several other nodes by a net. We will represent a net n_i as a set of the nodes that it connects. We denote the number of nodes in V by n the number of hyperedges in E by e , the average number of nets that a node is connected to by q_n , the average number of nodes that a net connect by q_e , and the average number of neighbors of a node by $d = q_n(q_e - 1)$ - a node u is said to be a neighbor of another node u , if u and u are connected by a common net. A k -way partitioning of G is a set of subsets $P^k = \{V_1, V_2, \dots, V_k\}$ of V such that each $u \in V$ belongs to exactly one V_i . Let r_1 and r_2 be two numbers between 0 and 1 such that $r_1 \leq r_2 \leq 1/k$ and $r_2 \geq 1/k$. Then, an (r_1, r_2) -balanced k -partition in which

$r_1 \leq |V_i| \leq r_2$ for each subset V_i of P^k . When $k = 2, r_1 = 1 - r_2$; however, for $k \geq 3$ there is no obvious relation between r_1 and r_2 (except $r_1 \leq r_2$). We assume that all nodes have unit size; the balance criterion is easily changed to reflect size constraints on the subset when this is not the case. The cutest of a k - way partitioning is defined as

$$E_{\text{cut}} = \{n_i \in E \mid \exists u_i \in n_i \text{ s.t. } u_i \in V_i, i \neq j\}$$

In other words, the cutest E_{cut} is the set of nets that connect nodes belonging to different subsets of P^k . The cost of size of the cutest of P^k is defined as

$$\text{cost}(P^k) = \sum_{i=1}^k c(n_i), \text{ where } n_i \in E_{\text{cut}}$$

Assume that there are n nodes in the hypergraph G , and that it is being partitioned into $\{V_1, V_2\}$. When partitioning a hypergraph or netlist, the gain of a node is not as apparent as in the case of graph. The FM netlist partitioner uses a simple extension of the Kernighan-Lin node gain calculation (used for graph partitioning). For each node u , let $I(u)$ be the set of nets to which u is connected that lie entirely in u 's current subset, and $E(u)$ be the set of nets that belong to the cutest and for which u is the only node connected to them in u 's partition. Then the gain of u is given by

$$\text{gain}(u) := \sum_{n_i \in E(u)} c(n_i) - \sum_{n_j \in I(u)} c(n_j)$$

This gain definition of a node is the immediate decrease in the cutest cost if it is moved to the other subset. The partitioning process proceeds by determining the next best node u_i to move in the i th step as follows. The “unlocked” node (initially all nodes are unlocked) with the maximum gain in either subset is determined. If the balance criterion on the two subsets can be maintained after moving this node from its current

subset to the other one, it is chosen as the node u_i . Node u_i is then moved to the other subset and “locked”, and the gain of all its neighbors are updated. The node gain (u_i) is inserted in an ordered set S , and the next best node is moved in a similar fashion such that the balance criterion is satisfied.

Data and analysis

We can represent this graph as an adjacency list.

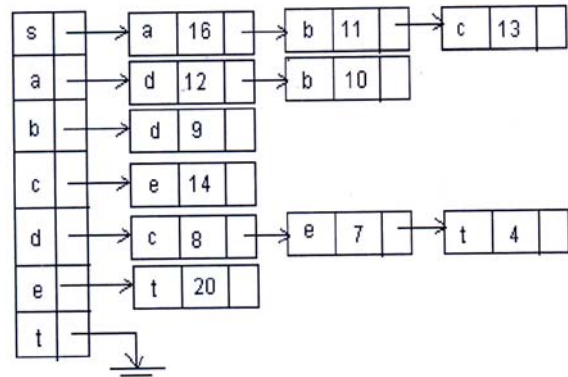


Figure 3. The Adjacency list representation of the graph

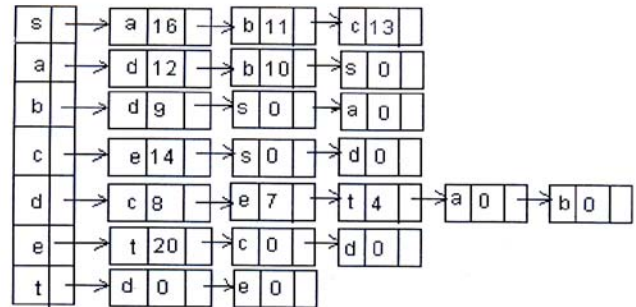


Figure 4. Adjacency list representation of G_f

From the adjacency matrix given in fig.3 we get the number of edges for the hypergraph connection different vertices in fig.4. The vertical and horizontal representation on the writing is analyzed through Cross-correlation from the result we find that hypergraph can be converted into ordinary spanning tree of a regular graph.

Step 1: Average of each row is subtracted from every entry d_1 .



Step 2: Average of each every column is subtracted from every entry d_2 .

Step 3: Product is summed and the average is taken.

Step 4: We find the cross correlation between this movements.

Row variance = 12.36

Column variance = 14.66

Covariance between row and column entities = 9.38

Correlation between row and column = 0.7

This shows that the hypergraph method is better than the vertical and horizontal method procedure in wing and cell partitioning. Markov distribution is shown below.

	1	2	3	4	5	6	7
1	0	.12	.14	.19	.14	.31	.06
2	.31	0	.28	0	.2	.21	0
3	.37	.17	.14	.11	.05	.08	.05
4	.15	.13	.18	.09	0	.26	.17
5	0	.14	.16	.21	.16	.32	0
6	.31	0	.28	0	.2	.2	0
7	.15	.12	.18	.09	0	.25	.18

Fig.5 Markov distribution

In fig.5 we have given the probability transition matrix for a design with finite number of nodes, but in VLSI design we have to deal with large scale of nodes. Thus this matrix will be of infinite dimension and the markov-chain is studied through virtual memory systems, where in additional memory locations are treated as virtual memories

5. CONCLUSION

In this paper, we have presented stochastic and deterministic models to tackle the challenges arising from the VLSI Circuit designs with wirelength optimization. Although significant progress has been in placement research, modern VLSI circuit designs have been induced many more challenges and opportunities for future research on partitioning and VLSI Cell placement.

REFERENCES

[1].R.C.Hilborn,Chaos and Quantization, (Cambridge University Press, New York

1994),pp.20-25.
 [2]. F.C.Moon,Chaotic and Fractal Dynamics: An Introduction Applied Scientists and Engineers (Wiley,NewYork 1992)
 [3] A.N.Kolmogorov., Dok.Acad. Nauk SSSR 119, pp 861 (1958) and Ya.G.Sinai,Doc.Acad.Nauk. SRRR 124,PP 768 ,(1958).
 [4]. M.Thiyagarajan,R.Manikandan, “A Novel Hybrid Stochastic VLSI Design Model” in Icfai University Journal of Computational Mathematics, Vol. II, No.4,December 2009, pp 46-55.
 [5]. A.Y.Hussein and Etwail,Convex Optimization and Utility Theory: New Trends in VLSI circuit layout. Ph.D Thesis ,Universityof Waterloo,Ontario,Canada,1999.
 [6].P.M.Pardalos, F. Rendl and H. Wolkowics, “The quadratic assignment problem: A Survey and recent developments” in DIMACS Series Discr. Math. Theor. Comp. Science, 1994, Vol.16, pp.1-42.
 [7].M.R. Garey and D.S. Johnson, Computers and Intractability, Freeman, San Francisco, 1979.
 [8]. A. Fritsch and O. Vornberger, “Cutting stock by iterated matching”, in Operations Research Proceedings, Selected Papers of the Int. Conf. on OR 94, U. Derigs, A. Bachem, and A. Drexl, Eds., Berlin, Springer. 1995, pp.92-97.
 [9]. Prof.M.Thiyagarajan and R.Manikandan “A Novel Graph Theoretical Algorithms for Constrained Placement Problems” International journal of Computational Intelligence and Information Security, Vol.1, January 2010, pp 26-33.
 [10]. Alpert, C. and Khang, K., 1996. “A hybrid multilevel/genetic approach for circuit partitioning”, Fifth ACM/SIGDA Physical Design Workshop, Reston, V.A., pp.100-105.
 [11]. Alpert, C.J., Huang, J.H. and Kahng, A.B., 1997. “Multilevel circuit partitioning”, Thirty fourth ACM/IEEE Design Automation Conf., Anaheim, C.A., pp.530-533.
 [12]. Bui, T. and Jones, C., 1993. “A heuristic for reducing fill in sparse matrix factorization”, Sixth SIAM Conf. Parallel Processing Scientific Computing, Norfolk, V.A., pp.445-452.
 [13].M.Thiyagarajan and R.Manikandan, “Hypergraph and Coprocessor design for VLSI Partitioning Problems”, Journal of Theoretical and Applied Information Technology, Vol. 26,No.2, pp 107-111,2011