

HYPER GRAPH AND COPROCESSOR DESIGN FOR VLSI PARTITIONING PROBLEMS

¹M.THIIYAGARAJAN, ²R.MANIKANDAN

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

² Assistant Professor, School of Computing, SASTRA University, Thanjavur, India -613401

ABSTRACT

VLSI Cell partitioning is considered as Hypergraph model, which can be treated as a randomized algorithm through the Markov chain. This approach helps to give a probabilistic algorithm through transition probability matrices of a Markov chain for VLSI partitioning. In the second model SAT problem situation is used to model FPGA Layout. As almost all problems posed in VLSI design and analysis are NP-Complete, any attempt to solve them is to identify some reduction tool to NP-Complete problems. One such reduction is attempted through SAT Problem.

Keywords: *Hyper Graph, Markov Chain, Max-Min Cut, Adjacency Matrix, SAT Problem, FPGA Layout, NP-Complete Reduction, Formal Language, Time Complexity.*

INTRODUCTION

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 relatively 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 [1], [2], [3].

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. The classification of nodes studied through an analysis on variety of classified states of the Markov chain fitted to the situation.

Section I:

1.1. 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 connects 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 $r_1 \leq |V_i|/n \leq r_2$ 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

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

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 user 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”

note (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.

1.2. Data and analysis

We can represent this graph as an adjacency list.

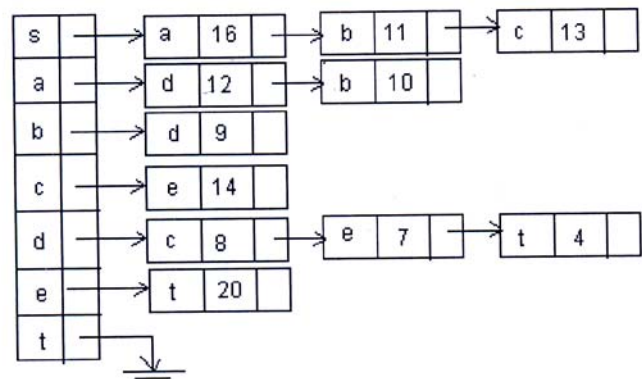


Figure 1. The Adjacency list representation of the graph

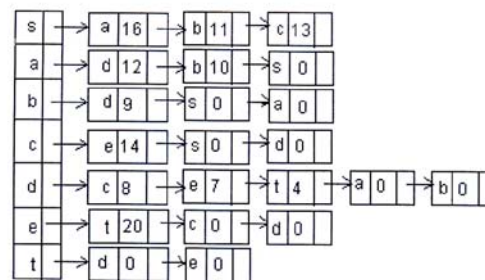


Figure 2. Adjacency list representation of G_f

From the adjacency matrix given in fig.3 we get the number of edges for the hypergraph connection different vertices in fig.5. 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 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.3 Markov distribution

In fig.3 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.

Section 2:

2.1. Boolean SAT based FPGA detailed routing formation [6]

In this approach geometric FPGA routing task is transformed into a Boolean satisfiability (SAT) equation with the property that any assignment of input variables that satisfies the

equation specifies a valid route. The satisfiability equation is then modeled as constraint satisfaction problem, which helps in reducing procedural programming. Satisfying assignment for particular route will result in a valid routing and absence of a satisfying assignment implies that the layout is unroutable.

2.2. FPGA layouts

One of the most commonly used layout models in FPGA applications [see Fig.4(a)]. It consists of two-dimensional array of configurable logic blocks CLBs, connection blocks C blocks and switching blocks S blocks. Each CLB marked L in [Fig. 4(a)] contains the combinational and sequential logic that implements the functionality of a circuit. C and S blocks contain programmable switch form the routing resources. C blocks connect CLB pins to channels via programmable switches. S blocks are surrounded by C blocks and allow signals either pass through.

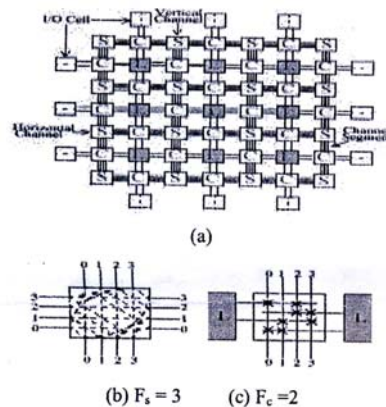


Fig.4. Island style FPGA model

A net consists of CLB pins that are interconnected with each other can be decomposed into one or more horizontal and/or vertical net segments each of which is an alternating sequence of C and S blocks forming an uninterrupted path. A detailed route of a net is a set of wire segments and routing switch following the topology specified by the global router such that no overlapping among detailed routes of different nets occurs.

The routing capacity of a given FPGA architecture is mainly expressed by three parameters W, F_c , F_s . The channel width is the number of tracks in a vertical or horizontal channel. The C block flexibility F_c is defined to be the number of tracks that each logic pin can connect to. The S block flexibility F_s denotes the number of

other tracks that each wire segment entering an S block.

Each wire segment entering this S block can connect to one track on each of the other three sides, hence $F_s = 3$ [See Fig.4(b)]. Each logic pin can be connected up to any two tracks in the C block, thus $F_c = 2$ [In Fig.4(c)].

The optimal assignment of tasks to processing is, in almost all practical cases on NP-complete problem. We must therefore make do with heuristics. These heuristics cannot guarantee that an allocation will be found that permits all tasks to be feasibly scheduled. We must check for feasibility.

These layouts can be considered the congruence classes in translating the NP-complete SAT problem to a polynomial time problem suggested by Hopcroft and Ullman [5]. Usually this problem is reduced to 3-SAT problem and the logic programming shown in fig.5 can be used to explain the situation.

SAT problem [5]

The Boolean expressions are unit from variables where values are Boolean (0, Y), logical operator AND and OR, any operator logical negation and parenthesis to great operators and operands.

A truth assignment of satisfies Boolean expression E assigns either true or false to each of the variables in E. A Boolean expression E is said to satisfiable if there exists atleast one truth assignment T that satisfied E. SAT problem is:

- given a Boolean expression is it satisfiable?

We have used to result the CSAT is NP-complete and in particular 3 SAT is NP complete.

We propose our layout problem as CSAT problem and used the construction of solution to solve our layout problem.

2.3. Data analysis

Here layout problem is considered and claim that work cover problem is SAT [5].

Proof:

The reduction is as follows. Let G with lower limit K be an instance of the independent-set problem. If G has n nodes, let G with upper limit n-

K be the instance of the node-cover problem. We construct, evidently this transformation can be accomplished in linear time. No claim that

G has an independent set of size K if and only if G has a node cover of size n-K.

(If) Let N be the set of nodes of G, and let C be the node cover of size n-K. We claim that N-C is an independent set. Suppose not, that is, there is a pair of nodes V and W in N-C that has an edge between them in G. Then since neither V nor W is in C, the edge (U, W) in G is not covered by the alleged node cover C. We have proved by contradiction that N-C is an independent set. Evidently, this set has K nodes, so this direction of the proof is complete.

(Only if) Suppose I is an independent set of C nodes. We claim that N-I is a node cover of with n-K nodes. Again, we proceed by contradiction. If there is some edge (U, W) not covered by N-I, then both V and W are in I, yet are connected by an edge, which contradicts the definition of an independent set.

Layout

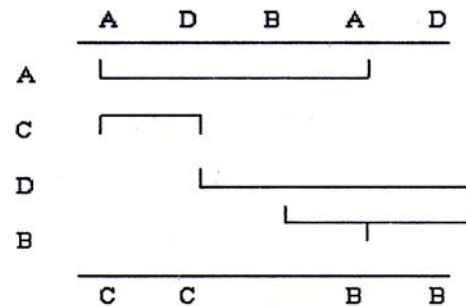


Fig.5a. Channel to be routed

Fig.5b. Matrix representation of Channel routing problem

$$E = (A \neq C) \wedge (A \neq D) \wedge (A \neq B) \wedge (C \neq D) \wedge (D \neq B)$$

Fig.5c. Exclusively Constraint

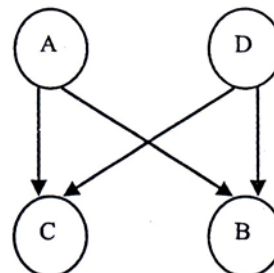


Fig.5d. Vertical Constraint Graph (VCB)

$$V = (A > C) \wedge (A > B) \wedge (D > C) \wedge (D > B)$$

Fig.5e. Vertical Ordering Constraint

$$R = E \wedge V$$

Fig.5f. Channel routability constraint

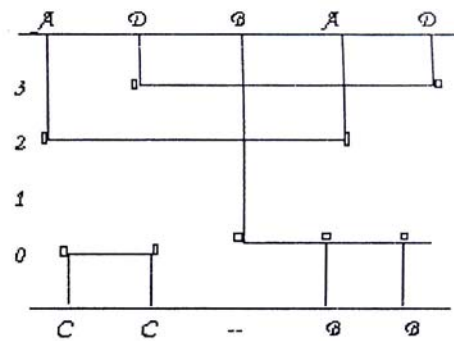
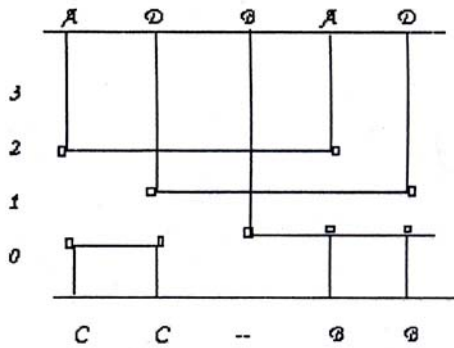


Fig.5g. Two feasible solutions

Fig.5. Boolean SAT modeling of channel routing problem

Any track number assignment that makes $R = 1$ corresponds to a feasible routing solution and completely specifies the net-to-track mapping. Two feasible assignments are shown in Fig.5g.

CONCLUSION

We have given an unified approach to VLSI partitioning problems considered so far over last two decades. A hybrid approach of Max-min algorithm, markov chain and hypergraph will give better result. An alternative method is considered through SAT problem . 3-CNF SAT problem is a convenient problem to reduce to other problems in order to solve them NP-Complete. Another NP-Complete problem that is often easy to reduce to other problems is the vertex cover problem. One

can use this vertex cover problem to any FPGA layout. Future work in this direction will help us to identify exactly other VLSI Design criteria.

REFERENCES:

- [1]. 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.
- [2]. 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.
- [3]. 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.
- [4]. Thiyagarajan, M., 2007. "Hypergraphs and Markov Chain", *International Journal of Computing and Application*, Volume 1 Number 2.
- [5]. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Velman, 2003. "Introduction to Automata Theory, Languages and Computation", Second Edition, Pearson Education, pp.452-453.
- [6]. Saveena, Vinay Chopra, Dr. Amardeep Singh, 2010. "Optimized FPGA Routing using soft-computing". *International Journal of Computer Applications*, Volume 7, No.8.