Low Cost Chip Set Selection Algorithm for Multi-way Partitioning of Digital System

Jae Young Park, Soongyu Kwon, Kyu Han Kim, Hyeong Geon Lee, and Jong Tae Kim

Abstract—This paper considers the problem of finding low cost chip set for a minimum cost partitioning of a large logic circuits. Chip sets are selected from a given library. Each chip in the library has a different price, area, and I/O pin. We propose a low cost chip set selection algorithm. Inputs to the algorithm are a netlist and a chip information in the library. Output is a list of chip sets satisfied with area and maximum partitioning number and it is sorted by cost. The algorithm finds the sorted list of chip sets from minimum cost to maximum cost. We used MCNC benchmark circuits for experiments. The experimental results show that all of chip sets found satisfy the multiple partitioning constraints.

Keywords—lowest cost chip set, MCNC benchmark, multi-way partitioning.

I. Introduction

AS VLSI integration is increased, circuit's complexity is growing fast. Problem of partitioning large circuit into smaller circuits has become very important. In this paper, we focus on partitioning problem to find chip set with lowest cost. Goals for the partitioning are to make chip's partition satisfy area constraint and minimize interconnection between each chip. The reason to minimize interconnection between chips is that if the number of interconnections is increased, numbers of required pins are increased in each chip. This increases the size of entire system and degrades system efficiency.

In this paper, partitioning problem is proposed when variety of chips of different sizes are given in a library. We focused on finding a chip set with lowest cost for multi-way partitioning.

In chapter 2, we define basic definitions. In chapter 3, we explain suggested algorithm to find most suitable chip set. Next, chapter 4 shows the analysis of the experimental results. Finally, conclusion is drawn in chapter 5.

II. Basic Definition

(Definition 1): k number of partitioning, $P^k = \{C_1, C_2, \dots, C_k\}$, is done by k's clusters, $C_1, C_2, \dots, C_k$. At that instant, it gets relationship of $C_1 \cup C_2 \cup \dots \cup C_k = V$. If k is 2, it becomes *Bipartition* problem.

(Definition 2) *signal net*, or *net*: A collection of pins which must be electrically connected [1].

Jae Young Park, Soongyu Kwon, Kyu Han Kim, Hyeong Geon Lee, and Jong Tae Kim are with the Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, South Korea(e-mail:jyp8389@gmail.com, sg.kwon@gmail.com, sgkh132@naver.com, dr2no3@hanmail.net, jtkim@skku.edu)

Fig. 1 is an example of partitioning. If a netlist of n module $V = \{v_1, v_2, \dots, v_n\}$ are given, partitioning is set by already defined k number of clusters that each module is assigned [2]. Minimized objective function is defined as $F(P^k)$ and this is partitioning result's cost function. Each cluster is assumed to be mutually exclusive.
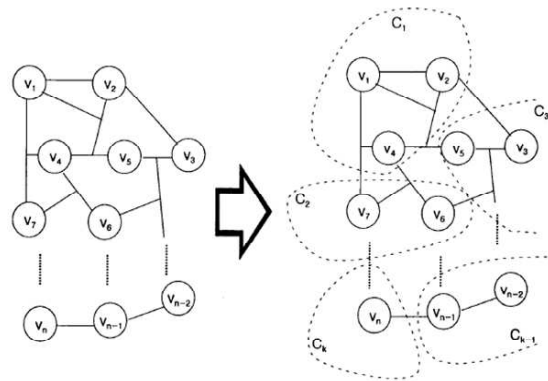

Fig. 1 Example of partition to cluster k

To express circuit's netlist in most regular way is hyper graph, $H(V, E)$. Here, $E = \{e_1, e_2, \dots, e_m\}$ is collection of signal net. Each e is collection of modules that connects this net. In all graphs, it is assumed that $e \in E$, $|e| \geq 2$. Weight function $w:V \rightarrow R$ is used in each module's area. If it is expanded to clusters, it becomes $W(C) = \Sigma_{v \in c} w(v)$ and another weight function, $w':E \rightarrow R$ can be defined by net.

(Definition 3): For each module v, nets that are adjacent to v are $N(v) = \{e \in E | v \in e\}$, and collection of modules neighboring v is expressed as $M(v) = \{w \in V | \exists e \ v, w \in e, v \neq w\}$.

(Definition 4): Each signal net e is created by one source module $S(e)$ and collection of destination modules $D(e)$. This allows knowledge of direction of the signal.

(Definition 5): Collection of Hyper-edge Cut due to cluster c is expressed as in next formula. $E(c) = \{e \in E \ s.t \ 0 < |e \cap C| < |e|\}$. Here, at minimum, if some of pins in net e are in cluster c, relationship, $e \in E(c)$, is formed.

III. Algorithm to Find Chip Set with Lowest Cost

The purpose of this paper is to find most suitable chip set before partitioning. The most suitable chip set means lowest cost chip set within maximum partition number. We consider chip's cost according to its size so we decide on each block's

size to materialize in a lowest cost. Next are necessary definitions.

*(Definition 6)*Chip set: Collection of chips to materialize some type of a system

*(Definition 7)*Maximum Partition Number: Chip's maximum possible number *(P_{max})* that makes up chip set

A. Problem Definition

Table I shows costs according to chip's area. As shown in Table I, as chip's area increases, cost also increases greatly. Especially, when area is increased from $1(cm^2)$ to $1.56(cm^2)$, die cost and packaging cost in greatly increased so that it becomes noticeable. System made with $1.56(cm^2)$ can be made with two $1(cm^2)$ chips so looking at TABLE I, system can be made with lower cost. If chip's cost is an important factor, it is better to get two smaller chips to comprise system with lower cost. Because of the reason, partitioning of integrate circuit is needed in many real cases.

TABLE I
COSTS FOR SEVERAL DIE SIZES[5]

Area (sq. cm)	Die /Wafer	Die yield /wafer	Cost of die	Cost to test die	Packaging costs
0.06	2778	79.72%	$0.25	$0.63	$5.25
0.25	656	57.60%	$1.46	$0.87	$5.25
0.56	274	36.86%	$5.45	$1.36	$5.25
1.00	143	22.50%	$17.09	$2.22	$5.25
1.56	84	13.71%	$47.76	$3.65	$52.25
2.25	53	8.52%	$121.80	$5.87	$52.25
3.06	35	5.45%	$288.34	$9.17	$52.25
4.00	23	3.60%	$664.25	$13.89	$52.25

Most of the previous partitioning algorithm has concentrated on heuristic method [3]. Partitioning algorithms using heuristic method need a proper initial solution, because heuristic methods start with temporary solution [4]. Having a better initial solution can make the final solution better. This is why we need a way to find chip set with lowest cost.

B. Most Suitable Chip Set

There can be many chip sets depending on number of types of chips and number of chips that consist chip set. According to Table II, if there can be 8 types of chips, number of chip sets available as much as maximum number of partition are combination with repetition. Here, *n* is maximum partition number.

As *n* gets larger, number of available combination increases. As *n* gets larger, system speed will get lower so n must be kept

TABLE II
NUMBER OF CHIPSET ACCORDING TO PARTITION NUMBER

Partitioning Number	Number of Chipset
N=1	8
N=2	36
N=3	120
N=4	330
N=5	792
N=6	1716
…	…

in reasonable value. This value made to be determined by the user's discretion. Next are assumptions for algorithms for selecting chip set.

In netlist, all cell's area is 1.
Connection area between each chips is ignored.
③ Chip's decrease in speed is ignored from partition

Each chip's factors, according to change in technology, like each chip's area, number of pins, and cost, are input from chip.tec file to get correct result. As situation changes, contents can be altered so we can get results from latest data. Maximum partition number is chosen by the user. This was done so that user can pick degree of system's decrease in speed due to partition. Fig. 2 shows actual *chip.tec* file.

```
5      /*Maximum Partition Number
8      /* Number of Chip types*/
/*Area, Pin Number, Cost*/
30        30       6
125       60       8
280       120      13
500       200      27
780       250      115
1125      300      199
1530      400      388
2000      500      811
```

Fig. 2 Example of *chip.tec* file

Using Fig. 2 chip.tec file, if comprised chip set where maximum area, where in TABLE II, if *n=5* and greater than *$20cm^2$*, of chipset is lower than input netlist, it is considered as case where it cannot find chip set that can satisfy area. In this case, new maximum partition number is input or output lowest cost chip sets with extent from lowest to maximum partition number.

C. Algorithm That Selects Most Suitable Chip Set

All chip set *($d_j=(A_j,P_j,U_j)$ $j=1,2,…,m$)* are put in array
– $D_{set}=\{d_1,d_2,d_3,…,d_m\}$
Eliminate all chip sets that has less area than input circuit's area *(A_{input})* from D_{set}.
Among remaining chip set *($d_j=\{ c_1,c_2,…,c_k\}$)*, where *k* is minimum, find lowest cost of chip set *(U_{min})*.
Eliminate all chip sets *(d_j)* from D_{set} where $U_{min}<U_j$.
All remaining chip sets in D_{set} are sorted by cost.
⑥ Select lowest cost chipset *(d_{min})* to be multi-way partition. When - $d_{min}=\{ c_1,c_2,…,c_k\}$), all chips get condition $c_j=(a_j,p_j,u_j)$, *($j=1,2,…,k$)*

While satisfying limitation of maximum partition number, to find lowest cost chip set, all chip sets that satisfies maximum partition number have to get cost calculated (first step). As shown in Fig. 3, this is done by pre-arranged array method according to chips with lowest area first to enumerate all chip sets *(D_{set})*. For all the chip sets that are acquired through this, we get area (A) and cost (U). In second step, chip sets that have smaller area than input netlist get ignored. In third step, we find

chip set that can materialize input netlist fastest that can use smallest partition. Its cost should be considered one of highest cost that can materialize input netlist. In fourth step, remaining chip sets are sorted by cost. Chip sets that are with higher cost than given cost are eliminated. Through this way, remaining chip sets are considered for materialization. In sixth step, chip set with lowest cost is applied multi-way partitioning algorithm to find partition that satisfies pin limitation. If it can't be found, next lowest chip set is applied multi-way partitioning algorithm.

```
0,0,0,0,0,0,0,0,0,1      //'0' means there is no chip
0,0,0,0,0,0,0,0,0,2      //maximum partition number is 10
0,0,0,0,0,0,0,0,0,3      //Each numbers mean chip type
0,0,0,0,0,0,0,0,0,4      //number in chip.tec file
0,0,0,0,0,0,0,0,0,5
0,0,0,0,0,0,0,0,0,6
......
8,8,8,8,8,8,8,8,8,6
8,8,8,8,8,8,8,8,8,7
8,8,8,8,8,8,8,8,8,8
```

Fig. 3 Enumerate all chip sets

IV. EXPERIMENT RESULTS

Program is written in C and circuit that was applied for this experiment is MCNC[93] Benchmark circuits. Table III shows circuits that were used in the experiment and cell number will be input circuit's area(A_{input}) because of assumption 1. chip's information is like Fig. 2.

Table IV is the *industry2* result. According to the algorithm to find chip set with lowest cost, we enumerate all chip set like Fig. 3 and eliminate all chip sets that has less area than idustry2 A_{input} area(12637). In third step, the chip set of U_{min} is {5,8,8,8,8,8,8} and k is 7 and the highest cost U_{min} value is 4981. After third step, $U_{min}<U_j$ is been eliminated and arranged by cost like Table IV. According to sixth step, the chip set {4,6,6,6,6,7,7,7,7,7} will be firstly selected to be partition.

V. CONCLUSION

TABLE III
MCNC BENCHMARK CIRCUITS

Circuit	Pin Number	Net Number	Cell Number
p1	2908	902	833
p2	11219	3029	3014
t2	6134	1720	1663
t3	5807	1618	1607
t4	5975	1658	1515
t5	10076	2750	2595
t6	6638	1641	1752
19ks	10547	3282	2844
Industry2	48158	13419	12637
S15850P	24712	10383	10470
S38584P	55203	20717	20995
S9234P	14065	5844	5866
structP	5471	1920	1952

TABLE IV
CHIP SET LIST OF INDUSTRY2 CIRCUIT

Order	Chip set	Partition Number	Total Area	Cost
1	{4,6,6,6,6,7,7,7,7,7}	10	12650	2763
2	{4,5,6,6,7,7,7,7,7,7}	10	12710	2868
3	{2,6,6,6,7,7,7,7,7,7}	10	12680	2933
4	{3,6,6,6,7,7,7,7,7,7}	10	12835	2938
5	{4,6,6,6,7,7,7,7,7,7}	10	13055	2952
6	{4,4,6,7,7,7,7,7,7,7}	10	12835	2969
7	{4,5,5,7,7,7,7,7,7,7}	10	12770	2973
8	{4,6,6,6,6,6,7,7,7,8}	10	12715	2997
9	{2,5,6,7,7,7,7,7,7,7}	10	12740	3038
....				
290	{3,5,5,5,7,7,7,8,8,8}	10	13210	3955
291	{1,7,7,7,7,7,7,8,8}	9	13210	3956
292	{2,7,7,7,7,7,7,8,8}	9	13305	3958
293	{6,6,7,7,7,7,7,8,8}	9	13900	3960
294	{1,1,7,7,7,7,7,7,8,8}	10	13240	3962
....				
	{5,8,8,8,8,8,8}	7	12780	4981

The purpose of the paper is finding low cost chip set for a minimum cost partitioning of a large logic circuits. Basically, partitioning of integrate circuit is needed in many real cases. Most of partitioning algorithm has concentrated on heuristic method. Partitioning algorithms using heuristic method need a proper initial solution and having a better initial solution can make the final solution better in heuristic method. This is why we proposed the algorithm to find low cost chip set.

We used MCNC benchmark circuits and implemented C program is comprised of algorithm that finds lowest cost chip set. In algorithm, input are netlist file and library file (*chip.tec*). User inputs currently possible type of chip information through library to let program get information of each chip's size, pin number, and cost in given library. Program would find chip set with lowest cost with given information. Output is a list of chip sets satisfied with area and maximum partitioning number. The output is sorted by cost according to the proposed algorithm from minimum cost to maximum cost. The experimental results show that all of chip sets found satisfy the multi-way partitioning constraints and the results can be used for multi-way partitioning.

REFERENCES

[1] S. M. Sait and H. Youssef, "VLSI PHYSICAL DESIGN AUTOMATION", IEEE PRESS, pp.28
[2] C. J. Alpert and A. B. Kahng, "Recent Directions in Netlist Partitioning: A survey" Integration : the VLSI Journal, 19(1-2), 1995, 1-81
[3] R. Kuznar, R. Brglez, and K. Kozminski, "Cost Minimization of Partitions into Multiple Devices", 30th ACM/IEEE Design Automation Conference.
[4] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by Simulated Annealing", Science, 220(4598), pp. 498–516, May 1983
[5] J. L. Hennessy and D. A. Patterson, "Computer Architecture A Quantitative Approach," , pp. 62