Thermal-driven Circuit Partitioning and Floorplanning with Power Optimization

Kyoung Keun Lee
School of Electrical and
Computer Engineering
Georgia Institute of Technology
777 Atlantic Drive NW
Atlanta, GA 30332-0250
leekk@ece.gatech.edu

Edward J. Paradise
School of Electrical and
Computer Engineering
Georgia Institute of Technology
777 Atlantic Drive NW
Atlanta, GA 30332-0250
gte737j@prism.gatech.edu

Sung Kyu Lim
School of Electrical and
Computer Engineering
Georgia Institute of Technology
777 Atlantic Drive NW
Atlanta, GA 30332-0250
404-894-0373
limsk@ece.gatech.edu

ABSTRACT

In this paper, we present methodology to distribute the temperature of gates evenly on a chip while simultaneously reducing the power consumption by using newly designed partitioning and floorplanning algorithms. This new partitioning algorithm is designed to partition blocks with well-balanced temperatures by altering the FM algorithm to include thermal constraints. Then, the suggested floorplanning algorithm can assign specific geometric locations to the blocks to refine the quality of the thermal distribution and to reduce power consumption. The combination of these two new algorithms, called TPO, is compared with the results of a conventional design procedure. As a result, power is reduced by up to 19% on average and a well-distributed thermal condition is achieved.

Keywords

Partitioning, floorplanning, Thermal distribution, Low power

1. INTRODUCTION

Increases in clock speeds and power consumption, plus denser chips, augment the importance of cutting-edge VLSI designs with strong thermal restrictions. The thermal properties of a chip directly influence the reliability. Unevenly distributed heat dissipation by gates in the chip may produce hot spots, which can reduce chip lifetime. To improve the reliability, many approaches have been presented. These approaches are based on floorplanning and placement algorithms [1, 2, 3, 4, 12]. Floorplanning algorithms have been used to reduce thermal reliability problems and even power consumption [1]. The matrix synthesis problem (MSP) models the thermal placement problem and suggests three algorithms to solve it [4]. A method to calculate temperature based on power estimation for standard cell placement has been introduced [12]. The standard cell thermal placement can be refined by a partitioning algorithm [3].

It is also important to estimate the power consumption of a chip. The estimated power consumption is used to calculate the temperature of gates on the chip. Switching activity based power estimation method has been presented [8, 10]. It has been introduced to estimate power consumption in large sequential circuits [9].

The smaller the variations in temperature across the chip, the more reliable it is. These thermal restrictions are best addressed early in the physical design of the chip. If these thermal issues are considered early in the design stage the optimized results of these design stages can be reserved throughout all other design stages. For example, the partitioning algorithm can reduce the global wires (= cutsize) among partitions. Then the reduction in the number of global wires is reserved in other stages – floorplanning stage and placement stage. This gives the main motivation of this paper.

Our project integrates these thermal restrictions right from the beginning in the partitioning phase. Along with the normal cutsize driven gain movements in the FM algorithm [7], thermal constraints are introduced which are tied in with pre-existing area constraints. These constraints limit the minimum and maximum temperatures of the blocks. Using the results from the partitioner, we further improve the thermal distribution of the blocks with suggested floorplanning algorithm. The A1 algorithm in [4] is used as an initial solution of the floorplanning algorithm. The results of the suggested algorithm are compared with the results of conventional approaches.

Our combined floorplanning and partitioning algorithms are called TPO. TPO showed reductions in power consumption for all our input files while maintaining and even thermal distribution. Gains of up to 19% in power reduction were observed.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation including definitions, and the thermal objective. In Section 3, the partitionng and the floorplanning algorithms are presented. Section 4 gives experimental results and future works are discussed in Section 5. Section 6 presents our conclusions.

2. PROBLEM FORMULATION

2.1 Definition and notation

Decomposition of the complex system into smaller subsystems is necessary for an efficient design. Each subsystem can be designed independently and simultaneously to speed up the design process. The process of decomposition is called *partitioning*. The system is called a *network*, which can be viewed as a set of gates or C cells $c_1, ..., c_C$ connected by a set of N nets $n_1, \ldots, n_N$. The set of cells are placed in partitions that are made by the partitioning. The cells have temperature values that depend on power consumption and thermal resistance [12]. The temperature of cells is defined as followings:

$$T_i = T_a + P_{total} \times R^{th} \cong T_a + \frac{P_{total}}{10^5 \times die\_size} \qquad (1)$$

where T_i is the internal chip temperature, T_a is the ambient temperature, P_{total} is the total power consumption of all cells, and R^{th} is the equivalent thermal resistance of the packaging components (°C/*W*). The temperature of partitions is defined as the summation of temperature of all cells in the partition. The temperature of partitions can be closed to each other by suggested partitioning algorithm with thermal constraints. It is called *thermal-balanced partitioning*.

In above equation, the thermal resistance values require the location information of cells. Therefore, the cells in the partitions need to get physical location information. The assigning of proper location is called *placement*. However, the partitions also need to get their own location information. This step is called floorplanning. By the proper floorplanning and placement, the cell's temperature can be distributed evenly. It is called *thermal-driven floorplanning and thermal placement problem*. The thermal placement problem can be represented as matrix structure problem (MSP) [4]. The MSP can also represent the thermal driven floorplanning with an assumption that all partitions have the same area and all cells in the partitions are located in the center position of the partitions.

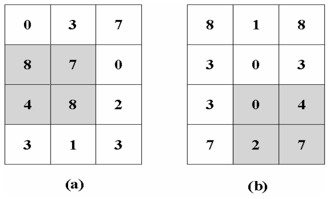
To solve the thermal-driven floorplanning problem, the *window temperature* is introduced to detect the change of thermal distribution by moving partitions or *blocks*. For example, a chip has 12 partitions, and the partitions have temperature values and are located arbitrarily in the Figure 1. In the Figure 1(a), the window on a chip is shown as a shadowed area and the window temperature is the summation of the temperature of all inside-blocks (= 27). This window has highest temperature among all windows. This window is called *hottest window*. Also, the *hottest block* can be defined as a block which has the highest temperature. On the contrary, the *coolest window and block* can be defined as the same way. To measure the quality of the thermal distribution, T_{MAX}, T_{MIN}, T_{Chip}, and standard deviation of the window temperature are used.

$$T_{MAX} = \max(W_t),\ T_{MIN} = \min(W_t),\ T_{Chip} = T_{MAX} - T_{MIN} \qquad (2)$$

where W_t is the temperature of window in a chip. The standard deviation of the temperature of windows can be defined as following:

$$S.D. = \sqrt{\frac{\sum (W_t - ave)^2}{total\#of\ windows}} \qquad (3)$$

where, *ave* is the average of the temperature of all windows.



0	3	7
8	7	0
4	8	2
3	1	3

(a)

8	1	8
3	0	3
3	0	4
7	2	7

(b)

Figure 1. An example of MSP [4] with 4x3 placement and 2x2 window. (a) window temperature is 27 (b) window temperature is 13.

2.2 Thermal Objective

The suggested partitioning algorithm makes thermal-balanced partitions with thermal constraints. The partitions are located in arbitrary position on a chip. The partitions can be relocated by suggested floorplanning algorithm to refine the thermal distribution on the chip.

To distribute heat dissipation evenly, the objective function of the first placement algorithms can be rewritten as

$$Obj = minimize\ T_{MAX} \qquad (4)$$

To achieve this objective function, the suggested floorplanning algorithm relocates the partitions iteratively. Another objective function is as following:

$$Obj = minimize\ S.D. \qquad (5)$$

2.3 Power Objective

2.3.1 Power consumption in Partitioning

The switching activity has an important role of dynamic power consumption in VLSI circuits. [8] proposed a method to get the switching activity in combinational and sequential circuits. The average dynamic power dissipation can be expressed as follows if the gate is the part of a synchronous digital system controlled by a global clock:

$$P_{avg} = 0.5 \times f \times V_{dd}^2 \times C_{load} \times E(transitions) \qquad (6)$$

where C_{load} is load capacitance, V_{dd}^2 is supply voltage, f is global clock frequency, and $E(transitions)$ is switching activities, which are the number of gate output transitions per clock cycle. Among these parameters, effective design factors to implement low-power design in physical level CAD design are capacitive load and switching activity in the dynamic power dissipation equation (6). The output load capacitance C_{load} in (6) consists of two elements: the capacitive load of a gate C_{gate}, which is the input and output capacitive load; and the capacitive load of interconnections among gates C_{wire}. Therefore, total dynamic power dissipation is given by

$$P_{avg} = 0.5 \times f \times V_{dd}^2 \times \sum_i (C_{gate}^i + C_{wire}^i) \times SA_i \qquad (7)$$

where SA_i indicated switching activity in a driving gate (=source) *i*. The capacitive load of interconnection is called wire capacitance C_{wire} that has two types of capacitive loads: local wire and global wire capacitive load. However, the information of the wires cannot represent real wire length in the partitioning step because there is no geometric information of the wires. Therefore, floorplanning and cell placement are used to get the geometric information.

In our algorithm, power optimization is performed in the floorplanning step. In floorplanning, the wire capacitance can be modeled as following:

$$C^i_{wire} = \alpha \cdot C^i_{gate} \cdot length(n_i) \quad (8)$$

where α is scaling factor and n_i is a net i which is connected between partitions. The length of nets inside partitions is ignored.

The objective function of the suggested floorplanning algorithm can be following equation:

$$Obj = \sum_{i\in\Phi} C^{wire}_i \cdot SA_i \quad (9)$$

where SA_i is the switching activity of a partition i, Φ is a set of partitions, and C^{wire}_i is the capacitive load of a global wire connected with partition i. The wire length of the global wire can be measured by Manhattan [5] distance between partitions.

3. ALGORITHMS

In this section, the suggested algorithms are explained in detail. The overview of the suggested methodology is shown in Figure 2. With temperature constraints, a newly designed FM partitioning algorithm can make partitions that have well-balanced temperatures. In the next step, the partitions are located on a chip arbitrarily. Then, the partitions are relocated by the suggested recursive floorplanning algorithm to refine the quality of the temperature distribution and power saving rate. The main benefit of the suggest method is to consider the thermal issue and power reduction feature in the earlier CAD design stage. That is, the optimized results of partitioning stage are reserved through all CAD design procedure. This overall algorithm is called the TPO algorithm.

3.1 Thermal-driven Partitioning

3.1.1 FM Overview

Our thermal-driven partitioner is based on the FM algorithm [7]. This iterative improvement algorithm focuses on reducing the number of cut nets during the partitioning process by calculating gains for each cell move from one partition to the other. These gains represent the reduction in number of nets that would be cut if the move were made. Both positive and negative gains exist for cells. The partitioning process begins with an initial solution where cells are evenly distributed between two blocks based on some area constraints. The area constraints are based on a user specified area skew that specifies the upper and lower number of cells the block can contain. From this initial solution, cells are then moved between the blocks. When all cells have been moved or the area constraints have been violated, a pass is completed.

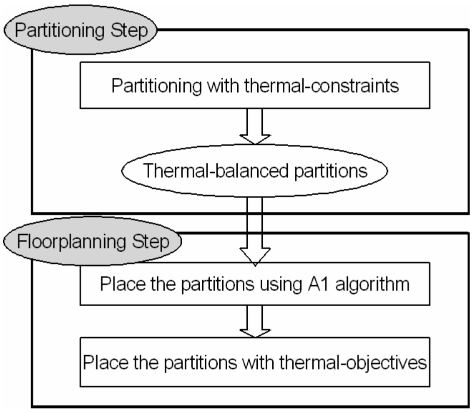

Partitioning Step
Partitioning with thermal-constraints
Thermal-balanced partitions
Floorplanning Step
Place the partitions using A1 algorithm
Place the partitions with thermal-objectives

Figure 2 Overview of the suggested algorithm (TPO).

The moves are then corrected to the point where the highest cumulative gain occurred and another pass starts.Passes stop when the cumulative gain is less than 1.

This algorithm can be utilized for more blocks than 2. In the simplest case, partitioning the first two blocks and then partitioning each of those blocks can make 4 blocks. The end result is 4 blocks of nearly equal size. The problem becomes slightly more difficult when the number of blocks is not a power of 2. In this case, the area constraints are not equal between the two blocks during the partitioning process. For example, if 3 blocks are desired, the area constraints for the first two blocks are a ratio of 2:1. The first block contains twice the number of cells as the second block. Then, the block with twice as many cells is partitioned once again and the final result is three blocks of nearly equal size. These area constraint ratios prove to be an interesting problem in determining the initial solution with thermal constraints.

3.1.2 Thermal Changes to FM

During the partitioning process thermal information was needed for each cell. The thermal information can be obtained by a method in [12]. However, since geometric information is unavailable during the partitioning process an exact temperature for each cell could not be obtained. In order to approximate the temperature of each cell the switching activity is used.

Two changes were made to the FM algorithm to incorporate thermal information. The first change is that the initial block setup must consider thermal constraints as well as area constraints. These constraints work exactly like the area constraints. Each block cannot be over an upper temperature or under a lower temperature. The second change to the algorithm is during the moves. Again, along with the area constraints thermal constraints are introduced in each move. The same thermal constraints that are used for the initial block setup are used during the cell moves. If a cell move violates one of the thermal constraints it is not accepted.

The initial setup of the blocks has now become more complicated, with a second set of constraints introduced. Before, the cells could even be separated into two blocks easily. Now, the same configuration of cells will very likely violate the thermal constraints, therefore a method must be introduced that satisfies both thermal and area constraints for the initial block configuration. This method must also take into consideration the area ratio between the blocks. If the ratio is 2:1, the thermal constraints must also reflect this so the temperatures of subsequent partitions will be evenly divided. Similar to area constraints, the user can specify a thermal skew. This skew represents how close the blocks should be in temperature. For example, if all the cells to be partitioned equaled a temperature of 100, a skew of 5 would mean that the temperature of a block would be between 45 and 55 inclusive.

The new method for thermal constraints is shown in Figure 3.

Incorporating Thermal Constraints

1. Find Thermal center values based on ratio of right block and left block (= lratio and rratio)
2. calculate total temperature of cells (= total_temp)
3. Set upper and lower temperature bound
4. Sort the cells based on switching activity
5. Insert cells into each of the blocks based on the ratios
6. If thermal constraints are violated by initial solution, swap the hottest cell from one block with the coldest cell from the other block until constraints are satisfied
7. Stop swapping if all cells have been swapped in one of the blocks

Figure 3 Procedure for incorporating thermal constraints.

In Figure 3, step 1 can be implemented by the following:

```
l_cen = lratio / (lratio + rratio);
r_cen = rratio / (lratio + rratio);
```

lratio is the ratio of left side and rratio is the ratio of right side in the bi-partition. In step 3, the following sets the upper and lower temperature bounds:

```
l_lower_temp = total_temp*(l_cen - temp_skew*.01 );
l_upper_temp = total_temp*(l_cen + temp_skew*.01 );
r_lower_temp = total_temp*(r_cen - temp_skew*.01 );
r_upper_temp = total_temp*(r_cen + temp_skew*.01 );
```

Total_temp is the total temperature of the cells and temp_skew is the specified temperature skew value. In step 5 the following code distributes the cells evenly based on area and temperature between the two blocks:

```
foreach cell in list:
      if(lratio > 0)
          add cell to left block;
          lratio --;
      else if (lratio == 0 and rratio > 0)
          add cell to right block;
          rratio --;
      if(lratio == 0 and rratio == 0)
          lratio = left ratio;
          rratio = right ratio;
end foreach
```

This initial block configure is a not trivial and deserves study on its own. This initial block setup does not work in all cases. When the number of cells in the current partition becomes small, one cell placement greatly affects the thermal configuration. For example, it was observed that in partitioning 40 cells into two equal blocks, the thermal constraints were [1.2, 1.3] yet there was one cell with the switching activity of 0.9. The thermal constraints and area constraints conflicted with each other in this method and a solution was unable to be found. In this case since the constraints are immediately violated, no moves can be made and the initial setup is accepted even though the constraints have been violated.

The additions to the move process are much simpler. When a move is considered, it is checked against the appropriate constraints for each block. If the cell is coming from the left block, first it is checked to make sure that the lower area constraint is not violated, and then the lower thermal constraint is checked. If the cell does now lower the area or temperature below the lower bounds, then the area and temperature of the right block is checked to make sure the cell does not increase the area or temperature of that block.

3.2 Floorplanning Algorithm

This section explains the refinement stage, which improves the quality of outputs. This suggested algorithm accepts the main idea of A1 algorithm in [4] to distribute temperature evenly. The A1 algorithm is designed to solve the placement problem, but TPO algorithm tries to adapt the main idea to solve the floorplanning problem.

First the A1 algorithm reorders the blocks by temperature. Next it makes groups that contain adjacent blocks of the same temperature. Finally, the A1 algorithm tries to replace the members of a group as far from each other as possible. The A1 algorithm performs well to distribute temperature evenly, but places the members of a group randomly. This means there is room to find a power reduction solution in the A1 algorithm's results while maintaining the temperature constraints.

This suggested algorithm tries to distribute temperature evenly using A1 algorithm first, then, reduce power consumption using the suggested TPO algorithm, which is trying to reduce the weighted wire length among blocks. The TPO algorithm finds the hottest block in the hottest window and the coolest block, then, it checks if swapping the two blocks increases T_{MAX}. If T_{MAX} is increased, the swapping is rejected. Among accepted swapping, choose only power saving results. The detail procedure for the floorplanning stage of TPO is shown in Figure 4.

Floorplanning Stage in TPO Algorithm

Input: Partitions with power and temperature information.
Output: Location of the partitions

1. Apply A1 algorithm and get initial solution
2. Find the hottest block in the hottest window among unlocked blocks.
3. Find the coolest block outside of the window among unlocked blocks.
4. Swap the two blocks.
5. If the T_{MAX} is increased, discard the swapping and go to 2. Otherwise, go to the next.
6. If the power consumption is increased, discard the swapping and go to 2. Otherwise, go to the next.
7. Update window temperature and the initial solution.
8. Go to 2 until all blocks are locked.
9. If there is improvement of power consumption, go to 2 with current best solution. Otherwise, stop a pass

Figure 4 Description of floorplanning in TPO algorithm.

Table 1 Partitioning results for the benchmark circuits with 64 blocks

			No thermal constraints			Thermal skew of 25			Thermal skew of 10			Thermal skew of 1		
circuits	cells	nets	cutsize	TB_{MAX}	TB_{SD}	cutsize	TB_{MAX}	TB_{SD}	cutsize	TB_{MAX}	TB_{SD}	cutsize	TB_{MAX}	TB_{SD}
misex3	3680	3666	824	96	3.58	861	83	3.01	1058	100	1.09	2647	86	0.23
seq	3005	2968	931	73	3.00	937	78	2.31	1023	72	1.24	2221	67	0.24
pdc	7582	7531	2038	199	4.31	3555	183	2.45	3843	191	0.84	6815	191	0.50
s13207	9517	8727	393	79.75	10.62	393	79.75	10.62	391	74.07	10.06	444	53.02	1.80
s15850	11081	10397	561	131.30	20.84	562	131.30	21.05	569	115.43	19.11	631	63.24	2.32
s35932	20164	18116	718	176.62	29.14	718	176.62	28.87	719	181.14	27.26	838	114.34	4.18
s38417	25803	24061	820	230.05	32.32	820	230.05	32.32	822	230.05	32.64	845	147.64	5.54
s38584	22601	20871	1169	214.80	31.27	1169	214.80	31.27	1172	214.80	30.93	1137	128.36	4.75

The partitioning step provides partitions as the input information of the suggested algorithm. The partitions are located by A1 algorithm (line 1). Then, TPO algorithm finds valid blocks to be swapped (line 2~4). If the block swapping satisfies given constraints, the swapping is accepted (line 5~6). Then, the TPO algorithm keeps finding candidates and swapping blocks to find better solutions (line 7~9).

The TPO algorithm can find better power saving solutions while maintaining the T_{MAX} value, because of the line 5 constraint in Figure 3. To satisfy the constraint, the coolest block is chosen as a candidate to be swapped (line 3). At this point, choosing only one candidate causes restrictions in finding a better solution. To improve the solution quality, it might be better to choose multiple candidates instead of choosing one candidate. However, choosing multiple candidates makes the algorithm more complicated and increases runtime dramatically. This more complicated algorithm makes new constraints that can reduce the possibility of choosing proper candidates. Therefore, the TPO algorithm accepts only one candidate as in line 3.

4. EXPERIMENTAL RESULTS

We implemented the suggested algorithms using C++ and the STL library in the UNIX environment. To measure the temperature value of cells, it is required to have the power dissipation of the cells and thermal resistance of a chip substrate. The power dissipation can be estimated by a power package in SIS [11]. Using the *power_estimate* command in SIS, the switching activities of gates were calculated by the symbolic simulation method [8] by the unit-delay model. SIS assumes primary input values were 0.5. However, we used a random pattern of 10,000 vectors to achieve a more realistic simulation instead of using the assumed primary input values. For a more valid simulation in this paper, we chose large benchmark circuits to compare out results with. However, we couldn't get the power information of these larger circuits because of the limitation of SIS power package. Therefore, SIS is used for only small sequential and combinational circuits and the power consumption of larger circuits are estimated by randomly generated switching activities that have an 80% portion in 0 ~ 0.5 range and a 20% portion in 0.5 ~ 1.0 range. These portions are based on the statistical analysis of the results of SIS. The capacitance value for each gate is determined by its fanout. Also, it is assumed the thermal resistance is 2000 K/W with 2 dimension substrate, and wire capacitance is 242*pF/m* [12]. The other parameters are identical with those in SIS.

From ISCAS89 and IWLS93 benchmarks, eight circuits in BLIF format [11] are used for comparison. The first five files are the largest circuits that we had (s38417, s38584, s35932, s15850, and s13207). These files have switching activity randomly generated and capacitance values based on fanout information. SIS is used to estimate power information for the last three files (pdc, seq, and misex).

4.1 Thermal Partitioning Results

The results from the partitioner proved promising. In Table 1, the results of the partitioner with 64 blocks and no thermal constraints are compared with results from the partitioner with thermal constraints. The values reported to compare these results are cutsize, TB_{MAX}, and TB_{SD}. TB_{MAX} is the maximum temperature among the blocks and TB_{SD} is the standard deviation of these same temperatures. The difference between the execution time of the original FM algorithm and our newly designed partitioner based on FM is negligible. The only added runtime is the swapping of cells between blocks when the initial constraints are violated.

Table 2 Change in cutsize and standard deviation for the benchmark circuits simulated with SIS.

Thermal skew	cut/orig_cut	orig_TB_{SD} / TB_{SD}
50	1.00	1.00
25	1.21	1.41
10	1.36	3.24
1	2.96	13.46

`

Table 2 shows the improvement in the standard deviation and deterioration of cutsize as the thermal constraints were tightened for circuits whose power information is from SIS.

In the randomly generated circuits, skew values of 50 and 25 did not affect the number of cells and nets were larger and were not affected by the thermal constraints until the skew was as tight as it can get. Small changes can be seen with a thermal skew of 10. The higher number of cells and nets allowed for a larger range of minimum and maximum thermal values. For example, the sum of 100 cells temperature if each cell was .5 would be 50. If there were 1000 cells, the temperature would be 500. If there was a thermal skew of 5, the range for two sets of circuits be [22.5, 27.5] and [225, 275] respectively. This larger

Table 3 Comparison results after floorplanning: window size is 2, the number of blocks is 256 and thermal skew of 10 for partioner.

ws 2	TPO / FM+A1				TPO / FLARE+A1			
256 way	TPow	VPow	S.D.	T_{MAX}	TPow	VPow	S.D.	T_{MAX}
misex3	0.96	0.95	0.46	0.65	0.98	0.98	0.39	0.55
seq	0.94	0.94	0.64	0.79	0.94	0.94	0.55	0.74
pdc	0.97	0.97	0.18	0.18	0.94	0.94	0.23	0.28
s13207	0.86	0.81	0.85	0.95	0.71	0.64	0.82	0.91
s15850	0.88	0.84	0.98	0.96	0.76	0.70	0.97	0.96
s35932	0.95	0.93	0.68	0.94	0.82	0.78	1.10	1.02
s38417	0.93	0.90	0.95	0.97	0.79	0.72	1.25	1.01
s38584	0.86	0.83	0.84	0.96	0.82	0.77	1.35	1.06
average	0.92	0.90	0.70	0.80	0.85	0.81	0.83	0.81

Table 4 Comparison results after floorplanning: window size is 4, the number of blocks is 256 and thermal skew of 10 for partitioner.

ws 4	TPO / FM+A1				TPO / FLARE+A1			
256 way	TPow	VPow	S.D.	T_{MAX}	TPow	VPow	S.D.	T_{MAX}
misex3	0.99	0.99	0.97	0.92	0.99	0.99	0.77	0.87
seq	0.94	0.94	1.60	1.03	0.95	0.94	1.21	0.99
pdc	1.02	1.02	0.18	0.48	0.99	0.99	0.28	0.63
s13207	0.86	0.81	1.26	0.99	0.68	0.59	1.09	0.98
s15850	0.93	0.91	1.49	0.99	0.81	0.76	1.50	0.99
s35932	0.94	0.92	0.94	0.97	0.81	0.77	1.53	0.99
s38417	0.92	0.89	1.08	0.98	0.78	0.70	1.39	0.99
s38584	0.88	0.85	0.87	0.97	0.86	0.83	1.60	0.99
average	0.93	0.92	1.05	0.92	0.86	0.82	1.17	0.93

range allows a propagation of higher temperatures down through the partitioning tree while still being within the thermal constraints for each block. The end result is that for larger circuits higher standard deviations are observed.

4.2 Floorplanning Results

The results of the TPO algorithm are compared with two conventional methods. In the first conventional method, the A1 algorithm is used as floorplanning algorithm with the partitioning results of the non-thermally-constrained FM algorithm. The second method uses A1 also as a floorplanning tool but it uses the partitioning results of FLARE, which is an advanced multi-level partitioning tool in terms of cutsize and delay [6]. The execution time of the floorplanning algorithm took a matter of seconds on a Sun E-450 server with 4 CPUs and 5GB RAN with multiple users running other applications.

Tables 3 and 4 both illustrate two different groups of ratios**.** The first group presents result ratio values of TPO compared to A1 paired with the original FM algorithm. The thermal skew of TPO is 10 in partitioning stage. This choice of a thermal skew of 10 is based on our experimental results that show tighter thermal constraints produce poorer power savings due to an increase in the cutsize. The second group displays ratio values of TPO compared to the A1 algorithm linked with the FLARE partitioner. The results in table 3 have 256 blocks with a window size of 2 and the results in table 4 have 256 blocks with a window size of 4 also. In both tables, TPow represents total power consumption that is calculated by eq. (7) and VPow represents visual power [13] consumption that is power consumption on wire. Also, standard deviation (S.D.) and T_{MAX} among window temperatures are calculated by eq. (2), (3). S.D. and T_{MAX} are used to measure the quality of thermal distribution. A ratio value of less than 1 indicates an improvement by TPO over these conventional methods.

The results show that on average both visual and total power are reduced. The maximum reduction on average in either visual or total power was 19% and the minimum reduction was 7%. T_{MAX} from the initial floorplanning solution is never increased after the floorplanning stage algorithm because TPO only accepts swaps where T_{MAX} is not increased (line 5 in figure 4). Therefore, almost all T_{MAX} ratios are below than 1, except for a few cases due to floating point truncation. Also, the standard deviation is only slightly increased. Therefore, the suggested algorithm can reduce power consumption while maintaining an even thermal distribution.

As well as observing the results of the thermal distribution and power consumption from the floorplanning stage the relationship between cutsize and power consumption can be inspected. We had expected that lowering cutsize would result in greater power reductions. This was the main motivation for the use of FLARE since its results have superior cutsizes over FM. However, it was observed that the A1 algorithm paired with FLARE has more power consumption than the A1 algorithm linked with the original FM algorithm. Therefore, although cutsize is still an important factor in reducing power consumption, it is not the only contributing factor.

5. FUTURE WORK

One interesting topic for further research is the addition of power reduction features in the partitioner, which would provide even better results in the floorplanning stage. Also, the initial block solution with thermal constraints could be further refined to produce solutions for all circuits regardless of the variations in switching activities. Finally, the current block swapping method could also be applied to the placement design stage.

6. CONCLUSIONS

Focusing on thermal constraints within the partitioning process proved an effective technique in both reducing the temperature of the hottest block from the partitioner and lowering the standard deviation in temperature among all the blocks. The tradeoff of reducing the temperature is an increase in cutsize. The block swapping strategy of the floorplanning algorithm reduces power while maintaining the thermal distribution from previous partitioning results. The combination of these two algorithms in TPO produced the desired results of an even thermal distribution and a reduction in power.

7. REFERENCES

[1] K. Y. Chao and D. F. Wong. Floorplanning for Low Power Designs. Proc. of ISCAS, pp. 45-48, 1995.

[2] K. Y. Chao and D. F. Wong. Thermal Placement for High-Performance Multichip Modules. Proc. Int. Conf. on Computer Design, pp. 218-223, 1995.

[3] G. Chen and S. Sapatnekar. Partition-Driven Standard Cell Thermal Placement. Proc. Int. Sym. Physical Design, pp. 75-80, Apr. 2003.

[4] C. N. Chu and D. F. Wong. Matrix Synthesis Approach to Thermal Placement. Proc. Int. Sym. on Physical Design, pp. 163-168, 1997.

[5] J. Cong and S. K. Lim. Physical Planning with Retiming. IEEE International Conference on Computer Aided Design, p2-7, 2000

[6] J. Cong and S. K. Lim. Performance Driven Multiway Partitioning. IEEE/ACM Asia South Pacific Design Automation Conference, p441-446, 2000

[7] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In Proc. 19th Design Automation Conf., pp. 175-181, June 1982.

[8] A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of average switching activity in combinational and sequential circuits. In Proc. 29th Design Automation Conf., 253-259, 1992.

[9] J. N. Kozhaya and F. N. Najm. Power estimation for large sequential circuits. IEEE Trans. on VLSI system, 9, 2, Apr., 2001

[10] F. N. Najm. A survey of power estimation techniques in VLSI circuits. IEEE Trans. on VLSI system, 2, 4, 446-454, Dec., 1994.

[11] E. M. Sentovich, *et. al.* SIS: A System for Sequential Circuit Synthesis. Memorandum UCB/ERL M92/41, UCB, Univ. of California, Berkeley, May 1992.

[12] C. H. Tsai and S. M. Kang. Standard Cell Placement for Even On-chip Thermal Distribution. Proc. Int. Sym. on Physical Design, pp. 179-184, 1999.

[13] H. Vaishnav and M. Pedram. Delay-optimal clustering targeting low-power VLSI circuits. IEEE Trans. On Computer-Aided Design, 18, 6, 799-812, June 1999.