Thermal-Aware High-level Synthesis Based on Network Flow Method

Pilok Lim, Taewhan Kim
School of Electrical Enginnering and Computer Science, Seoul National University
Seoul, Korea
polim@ssl.snu.ac.kr, tkim@ssl.snu.ac.kr

ABSTRACT

Lowering down the chip temperature is becoming one of the important design considerations, since temperature adversely and seriously affects many of design qualities, such as reliability, performance and leakage power of chip, and also increases the packaging cost. In this work, we address a new problem of thermal-aware module binding in high-level synthesis, in which the objective is to minimize the *peak* temperature of the chip. The two key contributions are (1) to solve the binding problem with the primary objective of minimizing the 'peak' switched capacitance of modules and the secondary objective of minimizing the 'total' switched capacitance of modules and (2) to control the switched capacitances with respect to the floorplan of modules in a way to minimize the 'peak' heat diffusion between modules. For (1), our proposed thermal-aware binding algorithm, called TA-b, formulates the thermal-aware binding problem into a problem of repeated utilization of network flow method, and solve it effectively. For (2), TA-b is extended, called TA-bf, to take into account a *floorplan* information, if exists, of modules to be practically effective. From experiments using a set of benchmarks, it is shown that TA-bf is able to use 10.1°C and 11.8°C lower peak temperature on the average, compared to that of the conventional low-power and thermal-aware methods, which target to minimizing total switched capacitance only ([18]) and to minimizing peak switched capacitance only ([16]), respectively.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: Integrated Circuits—*Design Aids*

General Terms

Algorithms, Design, Experimentation

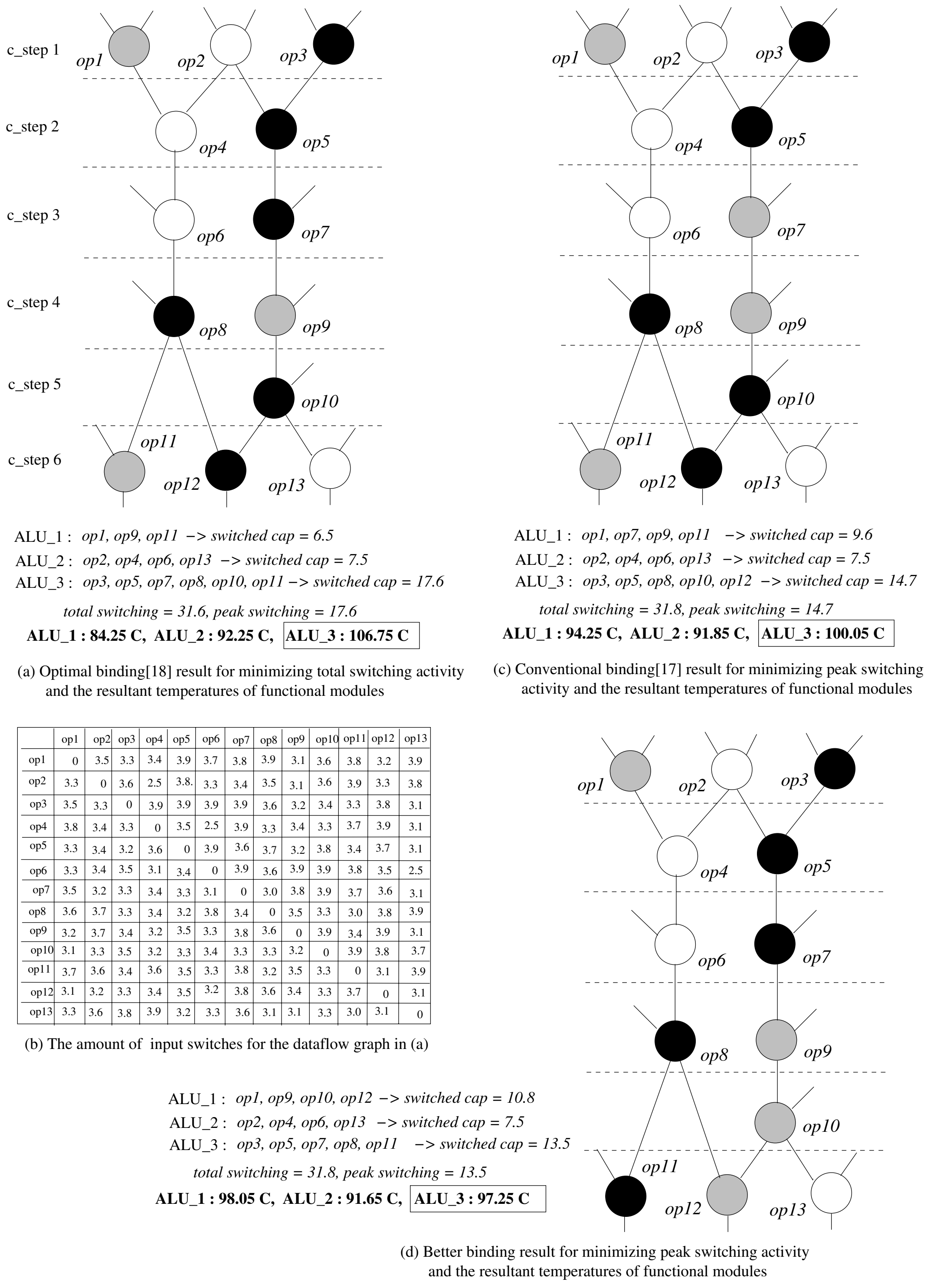
Keywords

Binding, Temperature, Power Consumption

1. INTRODUCTION

Due to the continued scaling of technology, designing with multimillion transistors is common. (For example, Intel Itanium 2 processor consists of over 200 million transistors [1].) However, the increase in the number of transistors in a limited silicon area causes a number of crucial design problems to be solved. One of them is power consumption due to the increase of power density in a chip. (For example, the power density in microprocessors would exceed that of a kitchen's hot plate at 0.6*um* technology [2].) One serious problem caused by the increase of power density is the increase of chip temperature. Furthermore, since different components of a chip can have different execution profiles, consuming different amount of powers, the temperatures of the components of chip are not uniform. That is, the component with heavy/light operating activity will reach a relatively high/low temperature. Consequently, in a designer's point of view it is very necessary to generate a chip in a way that the locations of the chip, which reach excessively high temperatures (called *hot spots*), do not appear.

Temperature has adverse effects in many respects. It reduces the chip's lifetime because of the acceleration of the chemical process (following Arrhenius equation) in the chip. It is shown that the mean time between failure (MTBF) of a chip is multiplied by a factor of 10 for every 30°C rise in the junction temperature [3]. Moreover, the increase of temperature decreases carrier mobility and thus switching speed of the transistors, which then increases the overall timing of the circuits. Also, temperature drastically increases the leakage power, which is becoming a major source of power consumption [4]. (For example, the leakage power can contribute as much as 42% of the total power in the 90nm process technology generation [5].) There has been a number of research works which have addressed the problem of thermal modeling and temperature-aware design (i.e, reducing the temperature of hotspots). For thermal modeling, a micro-architecture level thermal model was proposed in [6, 7], and a chip-level thermal model based on full-chip layout was proposed in [8]. On the other hand, in [9], a device level thermal modeling using heat transfer equations is proposed. In addition, the authors in [10] extended the thermal model in [6, 7] to be able to measure temperatures at different design granularities, and the authors in [11, 12] addressed the thermal modeling of interconnects. For thermal-aware design, most of the works focused on the physical level design. The authors in [13] developed a standard cell placement tool for the uniform thermal distribution on a chip. The authors in [14] proposed so called matrix synthesis approach to evenly place the cells with different thermal values. In addition, the authors in [15] proposed a thermal-driven floorplanning algorithm for 3-D ICs. Recently, the authors in [16, 17] proposed two module binding algorithms, one greedy and the other iterative improvement, for minimizing peak



ALU_1 : *op1, op9, op11 -> switched cap = 6.5*
ALU_2 : *op2, op4, op6, op13 -> switched cap = 7.5*
ALU_3 : *op3, op5, op7, op8, op10, op11 -> switched cap = 17.6*

total switching = 31.6, peak switching = 17.6

ALU_1 : 84.25 C, ALU_2 : 92.25 C, ALU_3 : 106.75 C

(a) Optimal binding[18] result for minimizing total switching activity and the resultant temperatures of functional modules

ALU_1 : *op1, op7, op9, op11 -> switched cap = 9.6*
ALU_2 : *op2, op4, op6, op13 -> switched cap = 7.5*
ALU_3 : *op3, op5, op8, op10, op12 -> switched cap = 14.7*

total switching = 31.8, peak switching = 14.7

ALU_1 : 94.25 C, ALU_2 : 91.85 C, ALU_3 : 100.05 C

(c) Conventional binding[17] result for minimizing peak switching activity and the resultant temperatures of functional modules

	op1	op2	op3	op4	op5	op6	op7	op8	op9	op10	op11	op12	op13
op1	0	3.5	3.3	3.4	3.9	3.7	3.8	3.9	3.1	3.6	3.8	3.2	3.9
op2	3.3	0	3.6	2.5	3.8.	3.3	3.4	3.5	3.1	3.6	3.9	3.3	3.8
op3	3.5	3.3	0	3.9	3.9	3.9	3.9	3.6	3.2	3.4	3.3	3.8	3.1
op4	3.8	3.4	3.3	0	3.5	2.5	3.9	3.3	3.4	3.3	3.7	3.9	3.1
op5	3.3	3.4	3.2	3.6	0	3.9	3.6	3.7	3.2	3.8	3.4	3.7	3.1
op6	3.3	3.4	3.5	3.1	3.4	0	3.9	3.6	3.9	3.9	3.8	3.5	2.5
op7	3.5	3.2	3.3	3.4	3.3	3.1	0	3.0	3.8	3.9	3.7	3.6	3.1
op8	3.6	3.7	3.3	3.4	3.2	3.8	3.4	0	3.5	3.3	3.0	3.8	3.9
op9	3.2	3.7	3.4	3.2	3.5	3.3	3.8	3.6	0	3.9	3.4	3.9	3.1
op10	3.1	3.3	3.5	3.2	3.3	3.4	3.3	3.3	3.2	0	3.9	3.8	3.7
op11	3.7	3.6	3.4	3.6	3.5	3.3	3.8	3.2	3.5	3.3	0	3.1	3.9
op12	3.1	3.2	3.3	3.4	3.5	3.2	3.8	3.6	3.4	3.3	3.7	0	3.1
op13	3.3	3.6	3.8	3.9	3.2	3.3	3.6	3.1	3.1	3.3	3.0	3.1	0

(b) The amount of input switches for the dataflow graph in (a)

ALU_1 : *op1, op9, op10, op12 -> switched cap = 10.8*
ALU_2 : *op2, op4, op6, op13 -> switched cap = 7.5*
ALU_3 : *op3, op5, op7, op8, op11 -> switched cap = 13.5*

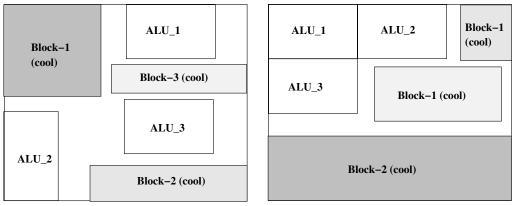
total switching = 31.8, peak switching = 13.5

ALU_1 : 98.05 C, ALU_2 : 91.65 C, ALU_3 : 97.25 C

(d) Better binding result for minimizing peak switching activity and the resultant temperatures of functional modules

Figure 1: Examples illustrating the effects of functional module binding on temperature.

temperature in high-level synthesis. Our proposed thermal-aware technique belongs to high-level synthesis. However, compare to the existing methods (e.g., [16, 17]), our approach entails the following two unique contributions: (1) *solving the thermal-aware module binding problem by utilizing a network flow based theoretical framework*, and (2) *solving the floorplan-driven thermal-aware high-level synthesis problem*. Note that contribution-1 is crucial to enhance the quality of the solutions, while contribution-2 is to reflect the heat transfer between the modules placed close to each other.

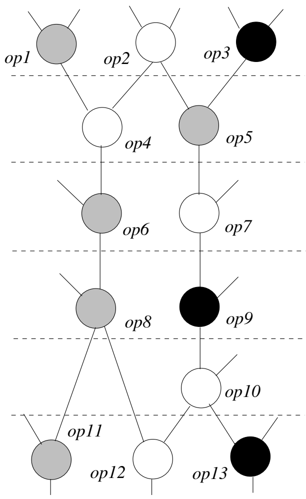


(a) Floorplan−1 which exhibits no heat flow between ALUs and other blocks

(b) Floorplan−2 which exhibits heat flow between ALUs and other blocks

	Results w/o considering placement information		Results considering placement information	
	sw cap.	temp.	sw cap.	temp.
ALU_1	20.7	127.05 C	11.6	106.05 C
ALU_2	22.1	117.05 C	22.1	110.15 C
ALU_3	13.0	90.05 C	23.4	115.25 C

(c) A comparison of results for floorplan−2 in (b) by minimizing the peak switching only, and by minimizing the peak switching considering the floorplan



ALU_1 : *op3, op9,op13* −> *switched cap =11.6*

ALU_2 : *op1, op5,op6, op8, op11* −> *switched cap =22.1*

ALU_3 : *op2, op4,op7, op10,op12* −> *switched cap =23.4*

total switching = 57.1, peak switching = 23.4

ALU_1 : 106.05 C, ALU_2 : 110.15 C, ALU_3 : 115.25C

(d) Floorplan−driven result for minimizing peak switching activity and the resultant temperatures of functional modules

Figure 2: Examples illustrating the effects of placement of functional modules on the temperature distribution.

2. MOTIVATING EXAMPLE

It has been observed and generally accepted that there are two major sources of heat generation in functional modules:
Observation-1: One source comes from the execution of modules. The amount of heat generated can be counted by the switched capacitances of modules. The switched capacitance of module m_i is proportional to its switching activity $sw(m_i)$ and its load capacitance C_L, i.e., $sw(m_i) \cdot C_L$. For simplifying the explanation, we assume, if not explicitly mentioned, that all load capacitance (C_L) values of modules are identical in the presentation of sections 2 and 3. Thus, minimizing the 'peak' switched capacitance among the modules corresponds to minimizing its (peak) switching activity.[1] Note that the experimentation section will explain how the C_L values are extracted and used.
Observation-2: Another heat source is from the neighboring modules in the placement. The modules which are placed closed to the modules with low (or high) temperature are likely to be cool (or hot).

Observation-1 implies that minimizing switching activity will be definitely helpful. However, minimizing the total amount of switching activity, sw_{tot}, of all modules does not always mean that the temperature of the hottest module goes down. For example, Fig. 1(a) shows a binding result using three modules, ALU_1, ALU_2, and ALU_3 for the scheduled dataflow graph. The binding result was obtained by using a power-optimal binding algorithm ([18]) for minimizing the quantity of sw_{tot}. (The hamming distance between the inputs of every pair of operations is given in the table of Fig. 1(b).) Consequently, sw_{tot} is a minimum, and the value is 31.6. The bottom of the dataflow graph in Fig. 1(a) summarizes the bounded operations to each module m_i and the amount of its switching activity $sw(m_i)$. With the obtained values of $sw(m_i)$ and the module placement in floorplan-1 in Fig. 2(a), we used the temperature estimator, HotSpot [6], to measure the temperatures of the functional modules. The computed peak temperature (T_{peak}) is 106.75°C on ALU_3, as summarized in Fig. 1(a). Also, we can see that the largest amount of switching activity i.e., peak switching sw_{peak}, also occurs at ALU_3. On the other hand, Fig. 1(c) shows the binding result produced by the algorithm in [17], which tries to reduce the value of sw_{peak} rather than that of sw_{tot}. Consequently, compared to that in Fig. 1(a), the resultant value of sw_{peak} is reduced from 17.6 to 14.7, which leads to the reduction of the value of T_{peak} from 106.75°C to 100.05°C, even the value of sw_{tot} is increased. This result implies that reducing sw_{peak} is more important than that of sw_{tot} to reduce the peak temperature, which means a more elaborated thermal-aware binding method is required. In fact, Fig. 1(d) shows a better binding than that in Fig. 1(c), which reduces the value of sw_{peak} from 14.7 to 13.5 and thus the value of T_{peak} from 100.5°C to 97.25°C.

Observation-2 implies that the information of module placement, if available, should be taken into account, which otherwise, the temperature distribution could be far from the expectation in the final implementation.

For example, suppose we have obtained a binding result of minimum peak switched capacitance using C_L values that are different from that used in Fig. 1(d), which, as previously shown, will be effective for the floorplans like floorplan-1 in Fig. 2(a) where no heat diffusion to other modules happens. However, consider another type of floorplan such as floorplan-2 in Fig. 2(b) in which the modules are placed together for the same binding result. Then,

[1] The 'peak' switched capacitance (switching activity) means the maximum of the switched capacitances (switching activities) of modules.

the resultant value of T_{peak} produced by HotSpot [6] is 127.05°C in ALU_1, as shown at the third column of the table in Fig. 2(c). Note that the module of sw_{peak} is ALU_2 but the hottest module is ALU_1. This is because as it can be seen in the placement of modules in floorplan-2, ALU_1 is surrounded by two hot modules, but ALU_2 and ALU_3 are neighbored with cool IP blocks.

On the other hand, Fig. 2(d) shows another binding, which considers the placement of the modules in floorplan-2. The result is summarized in the last two columns of the table in Fig. 2(c). We can see that ALU_3 has both sw_{peak} and T_{peak}, but the value (=115.25°C) of T_{peak} is much smaller than the peak temperature (=127.05°C) for the floorplan-unaware binding. This is because ALU_3 is placed in a location which allows the heat generated by the module to easily spread out. The observations imply that a careful thermal-aware binding technique is very necessary and further, the thermal-aware binding solution should take into account the placement information, if exists.

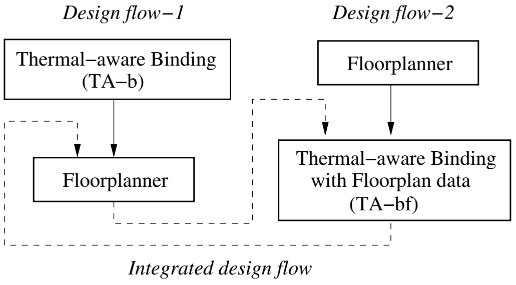

Design flow−1
Design flow−2
Thermal−aware Binding (TA−b)
Floorplanner
Floorplanner
Thermal−aware Binding with Floorplan data (TA−bf)
Integrated design flow

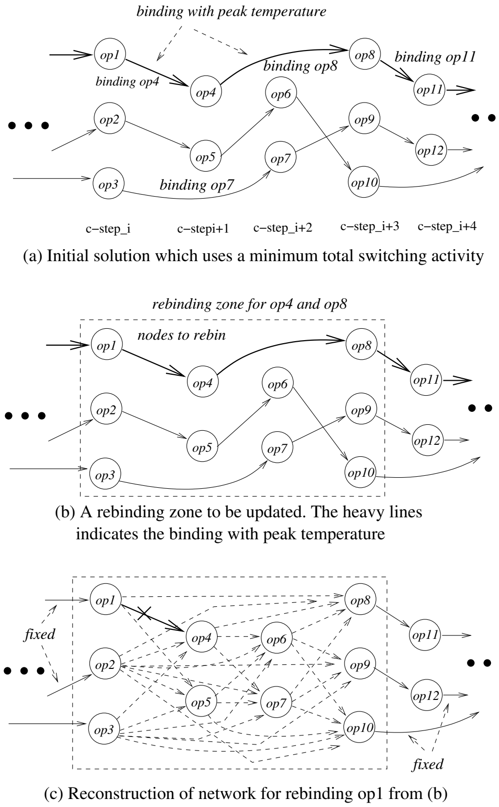
Figure 3: Design flows using TA-b and TA-bf and their integrated design flow.

3. THERMAL-AWARE HIGH-LEVEL SYNTHESIS

Our work consists of two parts: (1) a thermal-aware functional module binding algorithm TA-b, which is based on an effective utilization of the network flow method, and (2) floorplan-driven binding algorithm TA-bf, which takes into account the thermal effects of surrounding modules and/or IP blocks. Fig. 3 shows the two design flows where TA-b shall be used *before* floorplanning, as shown at the left side of Fig. 3, and TA-bf shall be used *after* floorplanning, as shown at the right side of Fig. 3. The peak temperature can be further reduced by combining the two flows, as indicated by the dotted arrows.

3.1 Thermal-aware module binding

When the floorplan information is not available, the best way TA-b can do is to minimize the maximum (i.e., sw_{peak}) switching activity of modules. (Here, a reasonable assumption we used is that the modules are completely isolated in the floorplan such that the heat diffusions between the modules never occur.) The proposed algorithm (TA-b) is an iterative improvement one. First, it generates an initial binding solution by formulating a low-power binding problem into a network flow problem (based on that in [18]), where the objective is to minimize the quantity of total switching activity sw_{tot}. For brevity, we omit details on how the graph for network flow formulation is constructed. (Fig. 4(a) shows a part of network in which the vertically arranged nodes correspond to the operations scheduled for execution at the same clock steps.) Rather we wish to conceptually show how the network is modified and updated[2] in



(a) Initial solution which uses a minimum total switching activity

(b) A rebinding zone to be updated. The heavy lines indicates the binding with peak temperature

(c) Reconstruction of network for rebinding op1 from (b)

Figure 4: An example to conceptually show how TA-b update the network.

our framework of thermal-aware binding algorithm TA-b to reduce the sw_{peak} value.

The basic idea of TA-b is, for each operation bound to the module of sw_{peak}, to attempt rebinding the operation by rerunning the network flow on an updated network. Then, among the rebindings, TA-b selects the one which results in the largest reduction on sw_{peak}, and performs the rebinding. TA-b repeats this rebinding process until there is no more reduction on sw_{peak}. Clearly, since the network flow method explores the search space globally, it is likely to find a solution close to an optimum. However, for large designs, the run time still takes long. To deal with large designs, we employ a network partition strategy: We divide the network into a set of multiple segments by cutting the network along the nodes corresponding to the 'full' clock steps, and apply network flow locally for rebinding. (A 'full' clock step is, if there are n functional modules available to use, the clock step at which exactly n operations are scheduled for execution. See Fig.4(b).) For example, let us assume that we have a segment of binding solution as shown in Fig. 4(a) from clock step-i to clock step-(i+4), where the solution produces a minimum value of sw_{tot}. The arrows indicate the flow paths (i.e., bindings) of the network. Let us assume that flow path $\cdots \rightarrow op1 \rightarrow op4 \rightarrow op8 \rightarrow op11 \rightarrow \cdots$ is a part of flow path of sw_{peak}. The dotted box in Fig. 4(b) shows a local *rebinding zone*, in which TA-b attempts two possible rebindings: one for $op4$ and another for $op8$. Fig. 4(c) shows the reconstruction of network which satisfies the constraint that $op4$ should never be bound to the same module to which $op1$ is bound. Note that the repeated applications of network flow computation reduces the sw_{peak} value. On the other hand, each application achieves a minimum sw_{tot} under the rebinding constraint. In that sense, it is well justified that TA-b reduces sw_{peak} as primary objective while trying to get down

[2] An example is shown in Fig. 4(c), and it will be explained later.

the value of sw_{tot} as much as possible. Furthermore, in the experiments, it is shown that a binding solution with a smaller value of sw_{tot} is more likely to be a smaller value of sw_{peak}.

3.2 Floorplan-driven thermal-aware high-level synthesis

Given a floorplan of modules, two critical factors that affect the temperatures of modules are: the switching activity of the module itself and the heat generated by the adjacent modules. Consequently, a proper control of the quantities of the two factors will lead to a considerable reduction on the peak temperature of the chip. The *heat diffusion* between two adjacent modules is proportional to their temperature difference and the length of the shared block boundary between them [21], expressed as:

$$H(m_i, m_j) = (T(m_i) - T(m_j)) \cdot L_{share}(m_i, m_j) \qquad (1)$$

where $H(m_i, m_j)$ is the amount of heat diffusion between modules m_i and m_j, $T(m_i)$ is the temperature of module m_i, and $L_{share}(m_i, m_j)$ is the length of shared boundary between m_i and m_j. A negative value of $H(m_i, m_j)$ indicates the heat flow from m_j to m_i. For a module m_l, the total heat diffusion of module m_l is defined by $H(m_l) = \sum_{\forall m_i} H(m_l, m_i)$ where m_i is adjacent to m_l. Then, *peak heat diffusion*, H_{peak}, is the maximum of $H(\cdot)$ values.

Note that the values of the two factors (i.e, the values of $sw(\cdot)$ and $H(\cdot)$) are tightly inter-related. The change of $sw(\cdot)$ values causes to change $H(\cdot)$ values, and conversely. Our strategy of minimizing T_{peak} is to gradually reduce the value of H_{peak} by adjusting the values of $sw(\cdot)$. Let F be the floorplan and B be the binding solution produced by TA-b. In addition, let ρ $(0 < \rho < 1)$ be a switching control parameter given by user. Then, the proposed thermal-aware binding algorithm integrated with floorplan information, called TA-bf, consists of three steps.

1. (*Computing heat diffusions*): For F and B, run a temperature estimation tool (e.g., [6]) and extract the temperatures of modules, from which we compute $H(\cdot)$ of each module.

2. (*Updating switching activity*): Let m_l be the module such that $H(m_l) = H_{peak}$. Apply TA-b to reduce $sw(m_l)$ as much as possible, but not more than $sw(m) \cdot (1 - \rho)$.[3] If $sw(m_l)$ is not reduced at all, stop and return B. Otherwise, i.e. $sw(m_l)$ is reduced to get down H_{peak}, we set B' to the new binding solution.

3. (*Controlling Temperature*): Recompute $H(\cdot)$ by extracting temperatures of modules for F and B'. If T_{peak} is reduced, set $B = B'$. Otherwise, i.e., T_{peak} is increased, set $\rho = \rho/2$. Repeat Step 2.

Step 1 is to compute the heat diffusions of all modules. In Step 2, we update the switching activity of each module by applying TA-b in such a way that $sw(\cdot)$ of H_{peak} is reduced by up to $100\rho\%$ (at maximum) while the total switching activity of all modules is minimized. In Step 3, we check if the change of switching activities reduces T_{peak} value. If T_{peak} is not reduced, a lighter change of switching activities is tried by decreasing the value of ρ by half. The procedure is terminated if TA-b, in Step 2, is no longer able to reduce the $sw(\cdot)$ value of T_{peak}.

4. EXPERIMENTAL RESULTS

The proposed peak temperature minimization algorithms TA-b and TA-bf were implemented in C++ and were executed on an Intel Pentium IV computer with clock speed of 1.5GHz. We tested benchmarks taken from various sources: applications from MediaBench [19], Discrete element methods [20], and high-level synthesis. (See the first column of Table 1.)

We extracted the dataflow graph for each benchmark, and simulated it to obtain the (averaged) switching activity for each pair of operations using a trace of 1,000 input values. The functional modules used were synthesized and optimized to logic netlist using Synopsys Design Compiler at 0.18um technology. The extracted capacitance value of each module and the switching activity on the module were used to compute the switched capacitance, from which we obtain the power trace.

We used HotSpot [6] to measure the temperature of modules. It takes as inputs the floorplan of modules, power trace of each module, and thermal model parameters. The thermal parameters are: $C_{convection} = 140.4J/K$, $R_{convection} = 0.1K/W$, Heat sink side = 6mm, Heat sink thickness = 6.9mm, Spreader side = 3mm, Spreader thickness = 1mm, Chip thickness = 0.5mm, Sampling interval = 3.33us. We assume the ambient temperature is 313.15 Kelvin (40°C).

We used the software developed by Goldberg ([22]) to solve the network flow formulations in TA-b and TA-bf as well as low-power network flow formulation ([18]).

Table 1: Temperatures of the hottest modules for the design produced by conventional power-aware binding, thermal-aware binding Seda and our TA-bf.

Benchmarks (#module)	Peak temperature (°C)			
	Power [18]	Seda [16]	TA-bf	red. [18]/[16]
EWF(2)	111.8	111.8	111.8	0.0/0.0
ADPCM_DEC(2)	141.3	141.3	141.3	0.0/0.0
MOTION_3D(3)	150.5	150.5	144.4	5.9/5.9
MOTION_3D2(3)	167.6	147.3	141.3	26.3/6.0
FORCE_3D(3)	124.9	119.8	107.5	17.4/12.3
FORCE_3D2(3)	173.6	230.9	152.6	21.0/78.3
MPEG_CAL2(3)	104.6	90.0	102.5	2.1/-12.5
MPEG_CAL3(3)	108.2	104.8	100.6	7.6/4.2
average				**10.1 / 11.8**

Two conventional methods Power ([18]), Seda ([16]) and TA-b are applied to the benchmarks and the results are summarized in Table 1. The first column indicates the tested designs and the number of functional modules allocated. Power is an optimal low-power binding algorithm which minimizes the total switched capacitance only. Seda is a thermal-aware binding algorithm, which minimizes the peak switched capacitance only. The difference between Seda and TA-b is that TA-b not only minimizes the peak switched capacitance but also minimizes the total switched capacitance. The floorplan we used here is the 'hotfloorplan' given by HotSpot. The floorplan is such that the temperature of each module is affected by the heat diffusion from the adjacent modules as well as the switched capacitance of the module itself. We see that for EWF and ADPCM_DEC, the peak temperatures by Power, Seda and TA-bf are the same. This is because the switched capacitances of the two modules are almost the same, and thus the heat diffusion between two modules was sufficiently enough to let the temperatures of modules to be identical. On the other hand, for FORCE_3D2, Seda produces the worst result. This is due to the unawareness of floorplan of modules in binding. One such case is that the module with peak switched capacitance assigned by Seda is the one sur-

[3] Instead of trying to rebind operations bound to the module with sw_{peak}, in this case TA-bf tries to rebind the operations to the module with H_{peak}.

rounded by multiple modules, causing a high heat diffusion. However, TA-bf will avoid such worst case by minimizing the peak heat diffusions. Overall, TA-bf uses 10.1°C and 11.8°C lower temperatures than that of Power and Seda, respectively. The comparison of results indicates that thermal-aware binding should take into account the floorplan information, if exist, and the idea used in TA-bf, which tries to minimize the peak heat diffusion, turns out to be effective.

5. CONCLUSIONS

In this work, we solved the thermal-aware module binding problem in high-level synthesis, in which the design goal is to minimize the *peak* temperature of the chip. This work entailed two contributions: (1) We formulated the problem into a problem of repeated utilization of network flow method, where we designed an optimization algorithm with the primary objective of minimizing the 'peak' switched capacitance and the secondary objective of minimizing the 'total' switched capacitance; (2) When a floorplan information was available, the switched capacitances to be reduced were 'selectively' minimized to reflect the degree of heat diffusions among the modules. From experiments with a set of benchmarks, it was confirmed that our proposed method was able to use 10.1°C and 11.8°C lower peak temperature on the average, compared to that of the conventional methods, which have solved the secondary objective only ([18]) and the primary objective only ([16]), respectively.

6. ACKNOWLEDGEMENTS

This research work has been supported by Nano IP/SoC Promotion Group of Seoul R&BD Program in 2006. This work was also partially supported by the Ministry of Science and Technology (MOST)/Korea Sceince and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc).

7. REFERENCES

[1] Intel *Itanium-R Processor Overview,* www.intel.com/design/itanium/

[2] S. Borkar, "Design challenges of technology scaling," *IEEE Micro,* 1999.

[3] National Semiconductor, *Understanding Integrated Circuit Package Power Capabilities,* www.national.com, April 2000.

[4] F. Fallah and M. Pedram, "Standby and active leakage current control and minimization of CMOS VLSI circuits," *IEICE Transactions on Electronics,* 2005.

[5] J. Kao, S Narendra, and A. Chandrakasan, "Subthreshold leakage modeling and reduction techniques," *ICCAD*, 2002.

[6] L. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," *ISCA*, 2003.

[7] L. Skadron, K. Sankaranarayanan, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture modeling and implementation," *ACM TACO*, 2004.

[8] T-Y. Wang and C. C-P. Chen, "3-D thermal-ADI: A linear-time chip level transient thermal simulator," *IEEE TCAD*, 2002.

[9] W. Batty *et al.*, ' "Global coupled EM-electrical thermal simulation and experimental validation for a spatial power combining MMIC array," *IEEE Transactions on Microwave Theory and Techniques,* 2002.

[10] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," *DAC*, 2004.

[11] K. Banerjee, *et al.*, "On thermal effects in deep sub-micron VLSI interconnects," *DAC,* 1999.

[12] T. Y. Chiang, K. Banerjee, and K. C. Saraswat, "Analytical thermal model for multilevel VLSI interconnects incorporating via effect," *IEEE Electron Device Letters,* 2002.

[13] C. Tsai, and S. Kang, "Standard cell placement for even on-chip thermal distribution," *ISPD*, 1999.

[14] C. C. N. Chu, and D. F. Wong. "A matrix synthesis approach to thermal placement," *ISPD,* 1997.

[15] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," *ICCAD,* 2004.

[16] R. Mukherjee, S. O. Memik, and G. Memik, "Temperature-aware resource allocation and binding in high-level synthesis," *DAC,* 2005.

[17] R. Mukherjee, S. O. Memik, and G. Memik, "Peak temperature control and leakage reduction during binding in high-level synthesis," *ISLPED,* 2005.

[18] J. M. Chang and M. Pedram, "Register allocation and binding for low power," *DAC,* 1995.

[19] C. Lee, M. Poktkonjak, and H. Mangione-Smith, "MediaBench: a tool for evaluating and synthesizing multimedia and communications systems," *ACM/IEEE International Symposium on Microarchitecture,* 1997.

[20] P. Cundal and O. Strack, "The discrete numerical model for granular assemblies," *Geotechnique,* Vol 29, pp. 1-8, 1979.

[21] Y. Han, I. Koren and C. A. Mortiz, "Temperature aware floorplanning," *Workshop on Temperature-Aware Computer Systems,* June 2005.

[22] A. V. Goldberg, "An efficient implementation of a scaling minimum-cost flow algorithm," *Journal of Algorithms,* Vol. 22, pp. 1-29, 1997.