Proceedings of the 2005 International Conference on Simulation and Modeling
V. Kachitvichyanukul, U. Purintrapiban, P. Utayopas, eds.

FLOORPLAN DESIGN FOR WIRING LENGTH MINIMIZATION IN ULSI CHIP USING SIMULATED ANNEALING

Apichat Terapasirdsin and Naruemon Wattanapongsakorn

Department of Computer Engineering
Faculty of Engineering
King Mongkut's University of Technology Thonburi
91 Prachauthit Road, Bangmod, Thung Kharu, Bangkok 10140, Thailand

ABSTRACT

Floorplan and placement are very critical problems in ULSI CAD design. The common ULSI system design objectives are such as improving or maximizing wiring ability, reliability and/or yield, and reducing or minimizing chip size, power consumption, crosstalk and coupling noise. These are crucial factors in designing next generation EDA (Electronic Design Automation) tools. Semiconductor scaling limits are forcing designers to look to novel circuits and design techniques to reduce design boundaries to maintain performance growth. In this research paper, we consider two system design objectives which are wiring length and crosstalk minimization. These are needed for greater modeling complexity and accuracy in EDA (Electronic Design Automation) tools. We propose methodologies and directions for floorplan design automation tools to meet the challenges ahead with Simulated Annealing (SA) approach. This approach is one of the efficient heuristic search algorithms to effectively predict and optimize various design objectives.

1 INTRODUCTION

As the complexities of ULSI circuits increase, Ultra Large Scale Integration (ULSI) refers loosely to placing more than about one million circuit elements on a single chip. The line between VLSI and ULSI is vague. The importance of electronic design automation tools in virtually every aspect of ULSI circuit design is undeniable. When chips become larger and clock frequencies increase, on-chip interconnect gains increased importance. Issues like wire-congestion and route ability, crosstalk and coupling noise, transmission line behavior, power consumption, reliability and yield, and their interrelation are crucial factors in designing next generation CAD tools. Semiconductor scaling limits are forcing designers to look to novel circuit and design techniques and to reduce design boundaries to maintain performance growth, all of which need for greater modeling complexity and accuracy in CAD tools.

Floorplan and placement are critical problems in ULSI CAD design. There has been progress in design techniques for the last few decades. However recent study on the optimal of current design tools shows the huge gap between current placement solution and the optimum solution (Chang, et. al. 2003), (Choi, et. al. 2003). It shows that the placement problem still requires a lot of research efforts as circuits continuously get larger and more complex. Placement for integrated circuits occurs after circuit elements have been mapped to specific physical circuit blocks or modules.

Our first objective is to minimize the length of the interconnections between modules with placement of floorplan design. In ULSI design, there is large number of interconnections which should be placed as close as possible to the associate modules. In order to reduce the total wiring length and obtain a high performance design. The placement problem should be computationally possible, simpler to calculate objective functions such as the area of the bounding rectangles, total wiring length interconnection (Koren, et. al. 1998), or some other routing area estimates. There is large number of interconnections. The placement problem with geometric constraints remains NP problem (Yang, et. al. 2002). For this reason, heuristic algorithms are commonly employed. These include Simulated Annealing (Sechen, 1988), (Mallela, et. al. 1988), force-directed placement (Sarrazedah, et. al. 1996) and Genetic Algorithms (Mazumdar, et. al. 1999). The most popular approach has been the iterative improvement algorithms, where each searching solution begins with a constructive step, but then uses an iterative component, repeatedly changing the solutions to the problem, in an attempt to get to the better solution. The Simulated Annealing (SA) remains the most popular iterative-

improvement approach in the literature (NCSU). There fore, we apply the SA algorithm in this research paper.

As ULSI fabrication technology evolves, coupling capacitance between adjacent wires is increasing rapidly over ground capacitance (Bakoglu, 1990). To further increase the operating speed, interconnection delay and transistor switching delay need to be kept as small as possible. However, large coupling capacitance not only degrades transistor switching delays, but also causes malfunction in the worst case. This is called crosstalk. Our second design objective is to minimize crosstalk (capacitance between adjacent wires) between channels with track assignment. The crosstalk problem is normally not considered during placement but it is considered after routing process. Floorplanning is the process of partitioning the entire chip area into smaller rectangles which will be occupied by various given building blocks. When considering on a floorplan design (Wimer, et. al. 1989), (Lengauer, et. al. 1990), we take into account the preferred common positions of the modules. This is determined by the number of nets that connect them. Two modules which have a large number of common nets should be placed as close as possible to reduce the total wiring length and obtain a high performance design by also reducing the crosstalk between the modules. Wiring length can be estimated from a given net metric, which is in the form of rectilinear (Manhattan) (Koren, et. al. 1998) distance. This metric contains distant values between the centers of the two modules which are connected by the given net. In this research, we consider row-based design concept, so that the crosstalk minimization process can be conveniently considered as the next design step.

The rest of this paper is organized as follows. The formulation of the problem is given in section 2. Section 3 describes our proposed algorithm for solving this problem. The experimental results are shown and discussed in Section 4, being following by the conclusion in Section 5.

2 PROBLEM FORMULATION

Floorplanning is the partitioning of the entire chip area into smaller rectangles which will be occupied by the various given building blocks. In the case of a microprocessor, these building blocks (also called modules) are such as instruction cache, data cache, instruction decode unit, integer arithmetic and logic unit, and alike. When deciding on a floorplan, we take into account the preferred mutual position of the modules, which is determined by the number of nets that connect them. Two modules which have a large number of common nets should be placed as close as possible in order to reduce the total wiring length and obtain a high performance design by reducing the communication delays among the modules. Since at the floorplanning phase the complete routing has not been attempted, only estimates of the wiring length can be taken into account.

2.1 Total number of wires connecting modules

A commonly used estimator for the wiring length of a given net is the rectilinear (Manhattan) distance between the centers of the two modules which are connected by the given net. Thus, the metric for the wiring length between two modules i and j which have M_{ij} nets in common, is given by

$$M_{ij}.(|x_i - x_j| + |y_i - y_j|) \quad (1)$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the centers of modules i and j, respectively. The total number of wires per module chosen so as to satisfy Rent's rule is

$$N_i = \sum_{i=1}^{n} M_{ij} + M_{i,out} \quad (2)$$

where M is the number of wires between modules i and j.

2.2 Wiring cost function

The total wiring cost for a given floorplan is designed as

$$W = \sum_{i=1}^{n}\sum_{j=1}^{m} M_{n_i n_j} D_{ij} \quad (3)$$

where D_{ij} is the Manhattan distance between modules i and j in a chip. Total of working area is

$$A_{total} = \sum_{i=1}^{n} h_i.w_i \quad (4)$$

where n is the number of modules. The floorplan is rearranged for the smallest wiring cost by fixing total width (w_r) of each row and adjusting height (h_r) of each row which are rearranged for wiring length minimization.

2.3 Simulated annealing

In this paper, we apply a simulated annealing algorithm (SA) (Sherwani, 1999) to find a sequence-pair which gives the best floorplan. In the SA algorithm, an initial solution is repeatedly improved by making small alterations until further improvements cannot be made by such alterations. Unlike greedy-type local search algorithms, the SA algorithm can avoid entrapment in a local minimum by allowing occasional uphill moves which deteriorate the objective function value. The uphill move is allowed with the probability given by exp (-Δ/T), where T is a control parameter called the temperature, and Δ is the difference between objective function values of the current and neighborhood solutions. The temperature is initially set with a certain method and gradually lowered in a predetermined method, called the cooling schedule. The

following shows how the SA algorithm is implemented for the floorplanning problem. The temperature T is initially set high. Therefore, the probability of accepting a move that increases the objective function is initially high. The temperature is gradually decreased as the search progresses. That is, the system is cooled slowly. At the end, the probability of accepting a move that decreases the objective function value becomes vanishingly small. In general, the temperature is lowered in accordance with an annealing schedule. The most commonly used annealing schedule is called exponential cooling, which begins at some initial temperature, T_0, and decreases the temperature in steps according to Ti+1 = αTi where 0 < α< 1. Typically, a fixed number of moves must be accepted at each temperature before proceeding to the next. The algorithm terminates either when the temperature reaches some final value, T_{final}, or when some other stopping criterion has been met.

3 FLOORPLANNING AND PLACEMENT ALGORITHMS

3.1 Floorplanning scheme

In this floorplan design step, we assume the area size of each module is 1 unit to find the minimal wiring cost pattern. This optimal pattern is used as a main guideline to construct the actual area in the placement process. After finishing in this step, a practical pattern will be occurred. A simulated annealing based on top-down multilevel partitioning placement flow used in Sun, et. al. 1993, Wang, et. al. 2000, Lin, et. al. 2002 is applied in our placement process. At every step of placement flow, the input circuits are recursively partitioned into rows in order to maintain minimal area size. Figure 1 shows the algorithm for cell moving.

We design floorplan with row based concept. Each module is arranged an inserted in a row as shown in Figure 1.

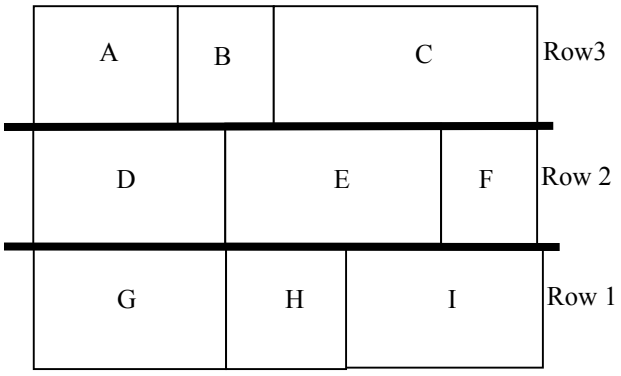

Figure 1: Floorplan of a Chip With N Modules

Total working area is

$$A_{total} = \sum_{i=1}^{n} Area_i \quad (5)$$

where *n* is the number of modules. The floorplan is rearranged for the smallest wiring cost by fixing total width (w_r) of each row and adjusting height (h_r) of each row which are rearranged for wiring length minimization. The shape of working area is assumed to be square. The total width (w_r) of each row is defined as (or else identified by a system designer).

$$w_r = \left\lfloor \sqrt{A_{total}} \right\rfloor \quad (6)$$

Each row is separated by a cutting which is a designed pattern for crosstalk minimization. At the first step of the procedure, the cooling temperature in SA algorithm is set to be very high where row exchange scheme is initially applied. Then the current solution is evaluated for its fitness value which is the total wiring cost, shown in Eq. (3). After the temperature is cooling down to a certain point, the module exchange scheme is randomly applied.

3.1.1 Interchange scheme

There are 2 patterns of interchange. We use temperature (T) to choose the interchange scheme. The first scheme is used when T is high. The second scheme is used when T is low.

1. Row exchange.

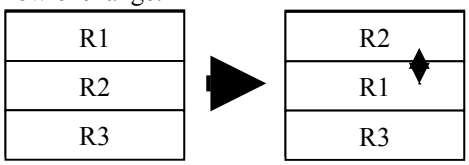

Figure 2: Row Exchange (Example: exchange R1 and R2)

2. Find a pair of rows that are separated in the y dimension in the given pattern. Find the pattern wiring cost using Eq.(3) and compare it with those of the original pattern.

3.1.2 Module exchange

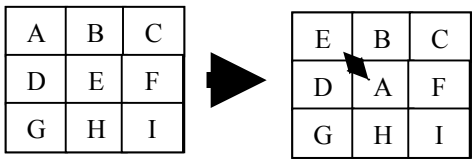

Figure 3: Modules exchange
(Example: exchange modules A and E)

Find new pattern by exchanging positions of the two selected modules. Find the wiring pattern cost using Eq. (3) and compare it with those of the original pattern.

In this step the total wiring cost can be reduced and may be closed to minimal when the temperature is decreasing. This process will terminate if the wiring cost does not change in certain iteration and the temperature is cooling down enough. For detailed description of the floorplan algorithm, see Figure 4.

1. For each pattern of module (x_i, y_i), compute the current wiring cost (W_{iter}); where iter = number of iteration.
2. Check temp
   If temp high : Swap row_i ←→row_j
   temp low : Swap Module(x_i,$y_{i)}$ ←→ Module(x_j,y_j)
3. Compare W if $W_{iter} > W_{iter+1}$: accept $pattern_{iter+1}$
   $W_{iter} < W_{iter+1}$: make decision with random x
4. Stop when termination criteria is reached and temp is low enough

Figure 4: Floorplan Algorithm

3.2 Placement Scheme

After we get the resulted pattern from the floorplan process, it cannot be directly used because the area of each module is not in practical dimension. Appropriate pattern can be obtained from placement process as shown in Figure 5 for building the actual size of each module. The exact wiring cost and actual area are then evaluated.

1. Input: area of each module, pattern, row number, and optimal pattern.
2. Calculated Row Width from Eq.(6) and Row Height (R_i)
3. Construct size of each module in row(in same row have same Height(Module Height in row_i = Row $Height_i$)
4. Find cost from Eq.(3)

Figure 5: Placement Algorithm

After rearranging modules with wiring length minimization objective, every cut between rearranged rows is considered for crosstalk minimization. This step is called channel routing, which will be conducted as our future work. Minimizing of the total wiring length and crosstalk significantly improve the performance of the chip.

4 A CASE STUDY & EXPERIMENTAL RESULTS

We implement our proposed approach and compare our result with those from existing research (Koren, et. al. 1998) and the result from an exhaustive search approach. We compared the wiring length after floorplan process.

Unlike other research works, we consider row-base approach for the wiring length minimization in order to ease the upcoming crosstalk minimization process, which will be reported in the near future.

In this example we examine a simple chip design consisting of nine modules (Koren, et. al. 1998)as shown in Figure 6 . All are soft modules that can be reshaped but they have to maintain their actual area size. These nine units compose of a ROM (R), static memory units, instruction cache and data cache, and random logic units with various transistor densities. We also assume the number of interconnects per module (which is n). Areas of each module are {A=4, B=4, C=2, D=8, E=4, F=2, G=6, H=3, I=3} and the wires connected from modules to external output pins are $M_{B,out}$ = 24 and $M_{G,out}$ = 32, which mean only modules B and G are wire-connected to external output pins..

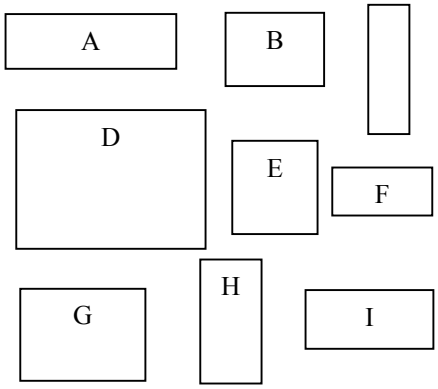

Figure 6: Nine Soft-Modules

In Figure 7, we assume that all modules have the same size for floorplan consideration. Then we compute the work area and row dimension.

A	B	C
D	E	F
G	H	I

Figure 7: Nine Modules With the Same Size (Each Module =1 Unit)

We consider the relation of each module with intermodule connection (M_{ij}) and the wires to the chip pins $M_{B,out}$ = 24 and $M_{G,out}$ = 32. and total number of wires per module was obtained from Eq.(2).

$$M_{ij} = \begin{array}{c} \begin{array}{ccccccccc} A & B & C & D & E & F & G & H & I \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 & 20 \\ 0 & 0 & 0 & 30 & 30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 30 & 0 & 0 & 0 & 15 & 30 & 30 \\ 0 & 0 & 30 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 0 & 15 & 15 & 15 & 0 & 15 & 10 \\ 24 & 0 & 30 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 20 & 30 & 0 & 0 & 0 & 10 & 0 & 0 \end{bmatrix} \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \end{array} \end{array}$$

We move the modules with interchange scheme and evaluate with cost function, Eq.(3), with relations of number of wiring between each module, (M[i,j]). The temperature is gradually decreased as the search progresses. That is, the system is cooled down slowly. At the end, the probability of accepting a move that increases W in Eq.(3) becomes vanishingly small. In general, the temperature is lowered in accordance with an annealing schedule. Exponential cooling begins at initial temperature,T0 = 100, and decreases the temperature in steps according to $Ti+1 = \alpha Ti$ where $0 < \alpha < 1$. When at high temperature ($T\_high \geq 80$), the temperature is decreased with Ti+1 = Ti - exp(T0)/n)), where n = number of modules. Otherwise at low temperature ($T\_low < 80$) is decreased with Ti+1 = Ti – 0.001. The algorithm terminates when there is no improvement in the solution's fitness value in a few iterations or other termination criterion has been met.

An optimal pattern obtained from our SA simulation is shown in Figure 8(b). This result is the same as the pattern obtained from exhaustive search approach and those from Koren , et. al.1998. However, we have a distinct approach with row-based concept for crosstalk minimization.

A	B	C
D	E	F
G	H	I

a) Original Scheme

A	B	F
H	I	G
E	C	D

b) Optimal Scheme
(From SA and Exhaustive Search)

Figure 8: Floorplan of a Chip With Nine Modules

From the SA simulation, we obtain the number of the wires to the chip pins $M_{B,out}$ = 21 and $M_{G.out}$ =32. The floorplan with minimal wiring cost is shown in Figure 8(b). the optimal layout is found with Tfinal = 79, iteration number = 173 and wiring cost of this pattern is 385 where the original pattern wiring cost is 653. Hence, the cost of original pattern is 58.95% higher than the cost of the optimal result.

After placement process, we apply the pattern in Figure 6(b) for constructing the dimension of each block with placement process. The result from SA is the optimal solution verified by an exhaustive search algorithm. This verification is possible when the problem size is not very large or the quantity of modules is relatively-small.. Detailed information of the placement outcome is shown below and graphically shown in Figure 9.

Total Area= 36 Area Width = 6 and Height= 6
Row_height R1 = 2.333333
Row_height R2 = 2.000000
Row_height R3 = 1.666667
Dimension:
At row height 1 =2.333333 Unit Width C = 0.857143
At row height 1 =2.333333 Unit Width D = 3.428571
At row height 1 =2.333333 Unit Width E = 1.714286
At row height 2 =2.000000 Unit Width G = 3.000000
At row height 2 =2.000000 Unit Width H = 1.500000
At row height 2 =2.000000 Unit Width I = 1.500000
At row height 3 =1.666667 Unit Width A= 2.400000
At row height 3 =1.666667 Unit Width B = 2.400000
At row height 3 =1.666667 Unit Width F = 1.200000
Position: (x, y):
Module A (0.000000, 4.333333)
Module B (2.400000, 4.333333)
Module C (1.714286, 0.000000)
Module D (2.571429, 0.000000)
Module E (0.000000, 0.000000)
Module F (4.800000, 4.333333)
Module G (3.000000, 2.333333)
Module H (0.000000, 2.333333)
Module I (1.500000, 2.333333)

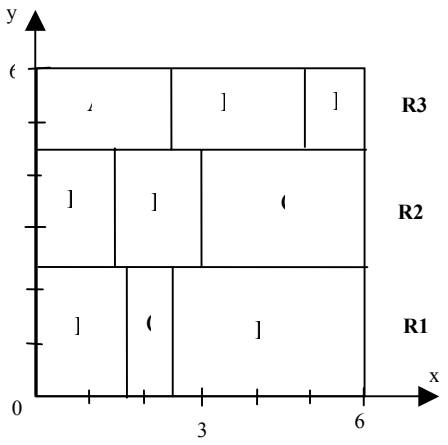

Figure 9: Floorplan With Nine Actual-Size Modules

Considering the actual size of all modules in the placement process, the minimum practical wiring cost is 667.428, which is 57.72% higher that the cost obtained previously from the floorplan process where each module is considered having one unit size. This resulted pattern of actual-size modules is shown in Figure 9, which will be later considered directly toward crosstalk minimization in our next research step.

5 CONCLUSION & DISCUSSION

The experimental results show that our SA algorithm is effectively moving cells to meet the design objective, where minimum wiring-length cost is obtained. We selected a small-size problem to demonstrate our design approach, and evaluate the SA result with the results obtained from a brute-force search technique. The results are confirmed to be optimal.

This floorplan & placement optimization problem is an NP-hard (non-polynomial) problem, which is difficult to solve when the problem size is large. In this case, traditional optimization approaches such as integer programming, branch and bound, and brute-force search algorithm cannot be used to find the optimal result because they require very high computational time and sometimes give invalid results. On the other hand, heuristic optimization approaches such as SAs are preferable because they give promising results with much less computational time.

As future work, we will perform crosstalk minimization using the result obtained from this floorplan & placement process, and consider if these two design objectives: wiring-length minimization (presented in this paper) and crosstalk minimization (including channel routing) can be considered simultaneously in the design process, creating a multi-objective problem solver.

REFERENCES

Choi, B.K., et. al. 2003. Floe-Based Cell Moving Algorithm for Cell Distribution. In *Proceedings of the 21st International Conference on Computer Design.*

Chang, C.C., J. Cong, and M. Xie. 2003. Optimality and Scalability Study of Existing Placement Algorithms. In *Proceedings of Asia and South Pacific Design Automation*: 621-627.

Kim, J.G., Y.D. Kim. 2003. A linear programming-based algorithm for floorplanning in VLSI design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 22:584 - 592

Cong, J., M. Romesis, and M. Xie. 2003. Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms. In *proceedings of International Symposium on Physical Design*: 88-94.

Linn, C.T., D.S. Chen, and Y.W. Wang. 2002. GPE: A New Representation for VLSI Floorplan Problem. In *Proceedings of the 2002 IEEE International Conference on Computer Design (ICCD'02)*.

Wang, M., X. Yang, and M. Sarrafzadeh. Dragon 2000: Fast Standard-cell Placement for Large Circuits. In *proceedings of International Conference on Computer-Aided Design*: 260-263.

Mazumdar, P., and E.M. Rudnick. 1999. *Genetic algorithms for VLSI Design, Layout and Test Automation*: Prentice Hall.

Sherwani, N. 1999. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers.

Koren I. and Z. Koren. 1998. Yield and Routing Objectives in Floorplanning. In *Proceedings of the 1998 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*: 28-36, November.

Sarrazedah, M., and C.K. Wang. 1996. *An Introduction to VLSI Physical Design.* McGraw-Hill.

W.I. Sun, and C. Sechen. 1993. Efficient and Effective Placement foe Very Large Circuits. In *proceedings of International Conference on Computer Aided Design*: 170-177

Lengauer, L. 1990. *Combinatorial Algorithms for Integrated Circuit Layout*: John Wiley&Sons.

Bakoglu, H.B. 1990. *Circuits, Interconnections, and Packaging for VLSI*. Addison Wesley.

Wimer, S., I. Koren and I Cederbaum. 1989. Optimal Aspect Ratios of Building Blocks. *VLSI, IEEE Transactions on Computer-Aided Design* 8: 139-145.

Mallela, S., and L. K. Grover. 1988. Clustering based simulated annealing for standard cell placement. In *proceedings of ACM/IEEE Design Automation Conference 25*: 312-317.

Sechen, C. 1988. Chip-planning, placement, and global routing of macro/custom cell integrated circuits using

simulated annealing. In *proceedings of ACM/IEEE Design Automation Conference 25*: 73-80

NCSU Collaborative Benchmark Laboratory. Available online via < www.cbl.ncsu.edu/pub/Benchmark_dirs/ LayoutSynth02 >[accessed October 01,2002]

Rexford, D., and J.D. Carothers. 2003. Cluster Growth Revisited: Fast, Mixed-signal Placement of Block and Gates. *Mixed-Signal Design, 2003. Southwest*: 243 - 248

AUTHOR BIOGRAPHIES

APICHAT TERAPASIRDSIN is a faculty member of Rajamangala Institute of Technology Northeastern Campus Nakhornratchasima Thailand. He received the B.S. degree in Industrial Education (Second Class Honors) (1997) and the M.S. degree in Computer Engineering (2000) from King Mongkut's University of Technology Thonburi (KMUTT). He is a Ph.D. candidate in Computer Engineering at King Mongkut's University of Technology Thonburi (KMUTT). His research interests include simulation and synthesis for VLSI, optimization algorithms, layout optimizations. His email address is <apichatter@yahoo.com>

NARUEMON WATTANAPONGSAKORN is an assistant professor in Computer Engineering at King Mongkut's University of Technology Thonburi. She received the B.S. degree and the M.S. degree, both from The George Washington University, and Ph.D. degree in Electrical Engineering from the University of Pittsburgh, USA. Her research interests include digital system design, network design and optimization, fault-tolerant technique, and reliability engineering. Her email address is <naruemon@cpe.kmutt .ac.th>