

COORDINATED SCIENCE LABORATORY

*College of Engineering
Applied Computation Theory*

**RECURSIVE
GATE-ARRAYS**

**C. Chiang
S. Maddila
M. Sarrafzadeh**

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS None	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-88-2262 (ACT#104)		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois	6b. OFFICE SYMBOL (if applicable) N/A	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research National Science Foundation	
6c. ADDRESS (City, State, and ZIP Code) 1101 W. Springfield Ave. Urbana, IL 61801		7b. ADDRESS (City, State, and ZIP Code) Arlington, VA 22217 1800 G. Street Washington, DC 20552	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Joint Services Electronics Program /NSF	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-84-C-0149 MTP-8709074	
8c. ADDRESS (City, State, and ZIP Code) Arlington, VA 22217 1800 G. Street, Washington, DC 20552		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) Recursive Gate-Arrays			
12. PERSONAL AUTHOR(S) Chiang, C.; Maddila, S.; Sarrafzadeh, M.			
13a. TYPE OF REPORT Technical	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) December 1988	15. PAGE COUNT 9
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>We propose a regular architecture, called <i>recursive gate-arrays</i>, suitable for circuits with modules of nonuniform size. A set of n (rectangular and L-shaped) modules can be placed in a recursive gate-array occupying $\Theta(A_{\min})$ area, where A_{\min} is the sum of area of the modules. The placement can be obtained in $O(n \log n)$ time.</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL		22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

Recursive Gate-Arrays¹

C. Chiang², S. Maddila³, M. Sarrafzadeh²

Abstract

We propose a regular architecture, called *recursive gate-arrays*, suitable for circuits with modules of non-uniform size. A set of n (rectangular and L-shaped) modules can be placed in a recursive gate-array occupying $\Theta(A_{\min})$ area, where A_{\min} is the sum of area of the modules. The placement can be obtained in $O(n \log n)$ time.

¹This work was supported in part by the National Science Foundation under Grant MIP-8709074 and ECS-8410902.

²Department of Electrical Engineering and Computer Science, The Technological Institute, Northwestern University, Evanston, IL 60208.

³Coordinated Science Laboratory and Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801.

Design automation is motivated by the complexity of present day computing systems in the VLSI environment. The use of a regular structure facilitates the design process. In gate-array architecture a square region of the plane is partitioned into equal-sized square subregions and each module (functional cell) is placed in a distinct region.

Consider a circuit μ which is a set of modules $\{M_1, \dots, M_n\}$, where n is assumed to be a perfect square. Let α_i denote the area of module M_i . A gate-array layout of μ requires $\Omega(n\alpha_{\max})$ area, where $\alpha_{\max} = \max_i \alpha_i$ is the area of the largest module of μ . Thus, gate-arrays are suitable for circuits with "equal-sized" modules [CFKNS]. For circuits with non-uniform size modules the gate-array layout may result in excessive wastage of the layout area. In this paper, we propose a recursive-gate-array structure which requires $\Theta(A_{\min})$ area to lay out an arbitrary circuit μ , consisting of rectangular and L-shaped modules, where $A_{\min} = \sum_i \alpha_i$.

In a *recursive-gate-array* (RGA) architecture, a square region of the plane is partitioned into equal-sized square subregions. Each square subregion either contains a module or is itself an RGA (see Figure 1).

Layout of a graph $G = (V, E)$, in the unit-square-grid $U = (P, L)$ (see [LP] for a formal definition of grids suitable for routing) is an embedding of G into U . Namely, each vertex $v \in V$ is mapped to a grid-point $p \in P$ and each edge $e \in E$ is realized by a sequence (l_1, \dots, l_k) of grid-edges ($l_i \in L$). We assume that each grid edge is used by at most one wire (i.e., using knock-knee layout mode where overlap of wires is not allowed [T]). Each module M_i is layout of a graph G_i ; boundary of each module is either rectangular or L-shaped. Area of each module is the area (measured in terms of grid units) enclosed by its boundary (see Figure 2).

If two modules correspond to the layout of the same graph then we say they can be *converted* to each other. If two modules with the same area (in an asymptotic sense) can

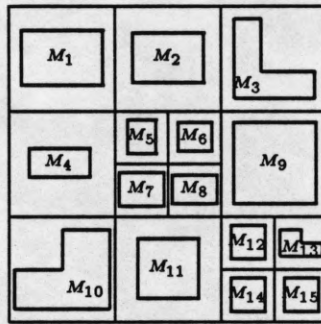


Figure 1: An RGA consisting of 15 modules

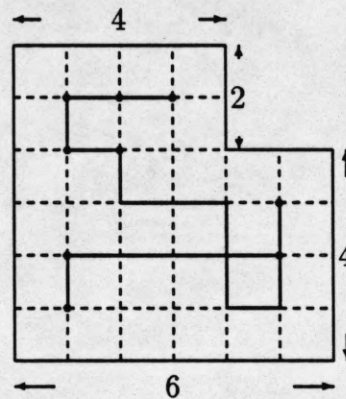


Figure 2: An L-shaped module with area 32

be converted to each other then they are called *equivalent*. Consider a module M occupying (grid) columns 1 thru ℓ . If we remove all (horizontal) wires between column i and column $i+1$ we obtain two modules: one with i columns and the other with $\ell-i$ columns. We say M has been *partitioned* along $x = i$. Hereafter, we write $M = (\omega, \ell)$ representing a rectangular module M with width ω and length ℓ ; by convention $\omega \leq \ell$. The next lemma shows a way o reconfigure rectangular modules without excessive wastage.

Lemma 1 *There exists a module $M' = (\omega', \ell')$ equivalent to a given module $M = (\omega, \ell)$, for $\omega \leq \omega' \leq \sqrt{\omega\ell}$.*

Proof: This result has been established in [L] for $\omega' = \sqrt{\omega\ell}$. Here, we prove the result for $\omega \leq \omega' \leq \sqrt{\omega\ell}$. Assume M occupies columns 1 thru ℓ . We partition M along columns

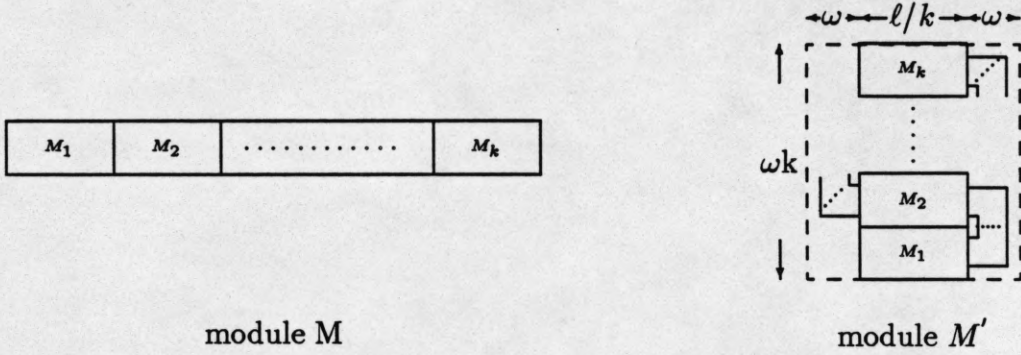


Figure 3: Converting a module

$x = \frac{\ell}{k}, x = 2\frac{\ell}{k}, \dots$, and $x = \frac{(k-1)\ell}{k}$ into k equal modules M_1, \dots, M_k , where $M_i = (\omega, \frac{\ell}{k})$ for all i . We place the southwest corner of M_i at the grid point $(1, \omega(i-1) + 1)$ and therefore its northeast corner at the grid point $(\frac{\ell}{k}, \omega i)$ (see Figure 3). Next we interconnect M_i and $M_{i+1}, 1 \leq i \leq k-1$, as they were connected in M , that is, we add wires to realize the removed wires. The module $M' = (\omega', \ell')$ is the rectangle enclosing all submodules and wires with $\omega' = \omega k$ and $\ell' = \frac{\ell}{k} + 2\omega$ (see Figure 3).

The area of M' is $\alpha' = \omega' \ell' = \omega k (\frac{\ell}{k} + 2\omega) = \alpha + 2\omega^2 k$, where α is the area of M . Next, we establish a bound on the extra area used by this layout. By hypothesis we know $\omega' = \omega k \leq \sqrt{\omega \ell} = \sqrt{\alpha}$ or $\omega k \leq \sqrt{\alpha}$. Since we have assumed $\omega \leq \ell$ (we have), $\omega \leq \sqrt{\alpha}$. By combining these two we get $\omega^2 k \leq \alpha$, or equivalently, $\alpha' \leq \alpha + 2\alpha$. Since M' and M correspond to the same graph and $\alpha' = \Theta(\alpha)$ then M' and M are equivalent. \square

Using the previous lemma we can change the aspect ratio (width/length) of a rectangular module. Our placement technique consists of two phases. In the first phase we convert all

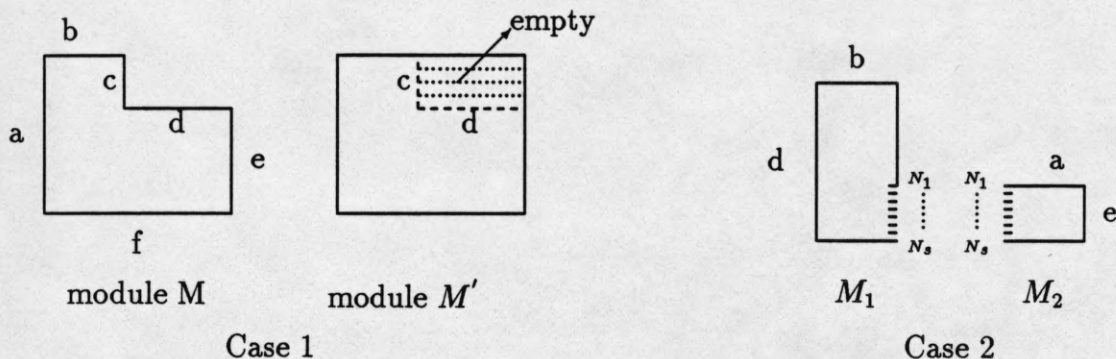


Figure 4: Converting an L-shaped module to a rectangular module

the modules into rectangles and in the second phase we reconfigure them appropriately.

Phase 1. Conversion to Rectangles

In this phase, all L-shaped modules are converted into rectangles. Let a, b, c, d, e , and f denote the six sides of an L-shaped module M in Figure 4, starting from the left side in a clockwise ordering. The area of M is denoted by α .

Case 1) length of side $e = \Theta(\sqrt{\alpha})$ and length of side $b = \Theta(\sqrt{\alpha})$

Since the area of M is $O(\alpha)$ (more precisely, α) then $|c| = O(\sqrt{\alpha})$ and $|d| = O(\sqrt{\alpha})$, where $|x|$ is length of side x . Let M' be the smallest rectangle enclosing M , and let α' denote its area (see Figure 4). Note $\alpha' = \alpha + |c||d| = \alpha + O(\alpha)$. Therefore, $\alpha' = O(\alpha)$.

Case 2) length of side $e = o(\sqrt{\alpha})$

We partition M (by cutting the nets N_1, \dots, N_s , where $s \leq |e|$) into two rectangles M_1 and M_2 , where $M_1 = (a, b)$ and $M_2 = (e, d)$. We denote areas of M_1 and M_2 by $\alpha_1 (= ab)$ and $\alpha_2 (= ed)$, respectively. (see Figure 4)

We convert both M_1 and M_2 into squares M'_1 and M'_2 of sides a_1 and a_2 , respectively (see Lemma 1). Without loss of generality, assume $a_1 \geq a_2$. Note that $a_1^2 = \Theta(\alpha)$. We place M'_2

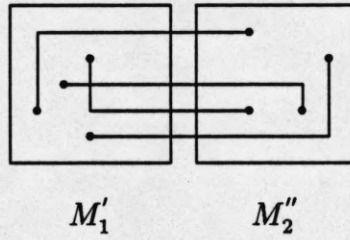


Figure 5: Interconnecting two sub-modules

in a square M''_2 of side a_1 . We place M'_1 and M''_2 , with total area $\Theta(\alpha)$, next to each other. Nets N_1, \dots, N_s are realized using s horizontal line and s vertical line. Thus each dimension is enlarged by $s = o(\sqrt{\alpha})$. (see Figure 5)

Case 3) length of side $b = o(\sqrt{\alpha})$

Analysis is symmetric to Case 2.

Thus we can write the following lemma.

Lemma 2 *An L-shaped module can be converted into an equivalent rectangular module.*

Phase 2. Placement

In this phase, first we convert all rectangles (output of Phase 1) into squares, as described in Lemma 1. We call the side of the smallest square one unit. Next we place each square in a square of side 2^i units, for some integer i . Note that the length of the side of the latter square is at most twice the length of the side of the former one, and thus, its area is at most four times the area of the former one. We obtain a set of square modules $\mu^* = \{M_1^*, \dots, M_n^*\}$, where α_i^* denote the area of M_i^* . By virtue of Lemma 1 and Lemma 2 we know $A_{\min}^* = \Theta(A_{\min})$, where $A_{\min}^* = \sum_i \alpha_i^*$.

We first place every four modules of side length 2^i in a module of side length 2^{i+1} . This

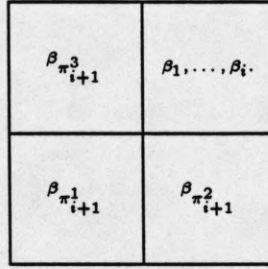


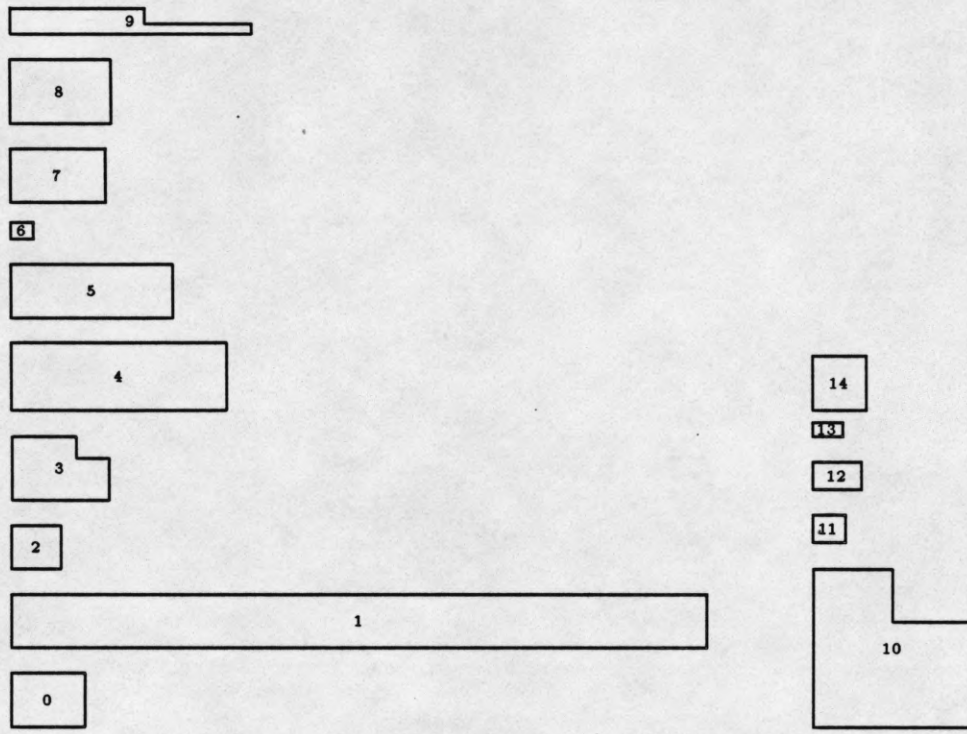
Figure 6: Recursive gate-array of $\beta_1, \dots, \beta_{i+1}$

process is continued till there are no more than three modules of each size. This gives a set of blocks B_1, \dots, B_k each of which is a recursive gate-array. Note that the sum of areas of all the blocks B_i is A_{\min}^* . Next, we sort all the blocks with respect to their area and obtain an ordering $\{(B_{\pi_1^1}, B_{\pi_1^2}, B_{\pi_1^3}), \dots, (B_{\pi_i^1}, B_{\pi_i^2}, B_{\pi_i^3})\}$, where groups of three blocks are all of the same size. Note that some of $B_{\pi_i^j}$ s may not exist. Let b_i be the side of $B_{\pi_i^j}$ s and $b_i < b_{i'}$ for $i < i'$. Let β_i denote the group of equal-sized blocks $(B_{\pi_i^1}, B_{\pi_i^2}, B_{\pi_i^3})$. Inductively assume that β_1, \dots, β_i have been placed in a recursive gate array of side $2b_i$. We can place $\beta_1, \dots, \beta_{i+1}$ in a recursive gate-array of side $2b_{i+1}$ as follows. Take a square of side $2b_{i+1}$ and partition it into four squares of side b_{i+1} . Place $B_{\pi_{i+1}^1}, B_{\pi_{i+1}^2}$, and $B_{\pi_{i+1}^3}$ in three of the squares and place the recursive gate-array corresponding to β_1, \dots, β_i , with side $2b_i$, in the fourth square. This is always possible since $2b_i \leq b_{i+1}$ (see Figure 6).

By induction, we conclude that β_1, \dots, β_t can be placed in a recursive gate-array of side $2b_t$. Since sum of the areas of B_i s is A_{\min}^* (or $\Theta(A_{\min})$), and is greater than b_t^2 (for there is at least one module with side b_t) then $(2b_t)^2 = \Theta(A_{\min})$. Note that each (conversion) step takes $O(1)$ time per module and sorting requires a total of $O(n \log n)$ time. We conclude:

Theorem -1 *A recursive gate-array layout of n modules, with minimum achievable area, can be obtained in $O(n \log n)$ time.*

The proposed placement technique has been implemented. An example is given in Figure 7.



input data

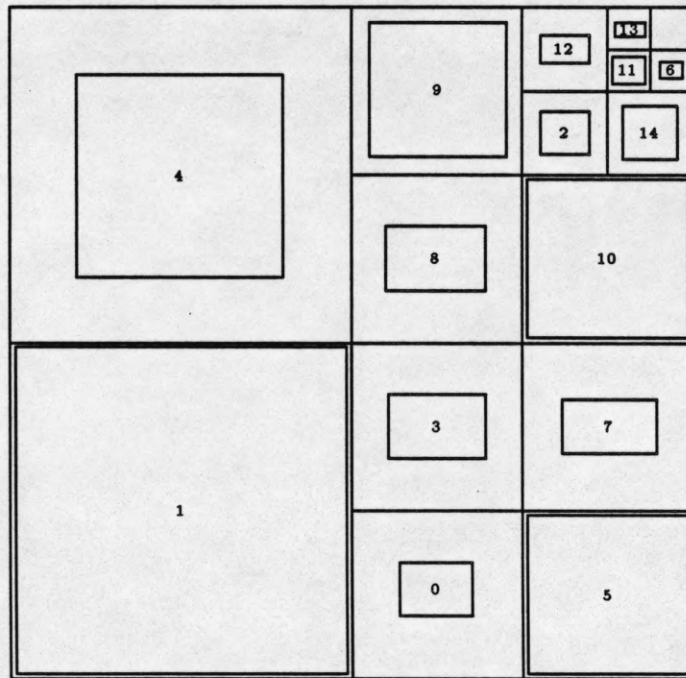


Figure 7: Demonstrating the proposed algorithm

References

- [CFKNS] K. Chen, M. Feuer, K. Khokhani, K. Nan, and S. Schmidt, "The Chip Layout Problem: An Automatic Wiring Procedure," *Proceedings of the 14th Design Automation Conference*, June 1977, pp. 298-302.
- [L] C. E. Leiserson, "Area-efficient Layouts for VLSI," *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, 1982.
- [LP] W. Lipski and F. Preparata, "An Elementary Theory of Wirability," *Mathematical Systems Theory*, Vol. 19, No. 3, 1987. pp. 189-204
- [T] C. D. Thompson, "A Complexity Theory for VLSI," Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, 1980.