Layout Generation of Array Cell for NMOS 4-phase Dynamic Logic

Makoto FURUIE† Bao-Yu SONG† Yukihiro YOSHIDA† Takao ONOYE†† Isao SHIRAKAWA†

† Dept. of Information Systems Engineering
Osaka University
Suita, Osaka, 565-0871 Japan
Tel: +81(6)6879-7808
Fax: +81(6)6875-5902
e-mail:{furuie,song,yoshida,sirakawa}@ise.eng.osaka-u.ac.jp

†† Dept. of Communications and Computer Engineering
Kyoto University
Kyoto, Kyoto, 606-8501 Japan
Tel: +81(75)753-4803
Fax: +81(75)753-4804
e-mail:onoye@kuee.kyoto-u.ac.jp

Abstract— An array cell (AC) architecture for the layout design is described, which is dedicated to low-power design by means of the NMOS 4-phase dynamic logic. An AC is constructed of (M×N)+2 transistors so as to constitute each type of NMOS 4-phase logic gates. A graph theoretic approach is exploited in the layout design to reduce the layout area. A number of experimental results demonstrate the practicability of the proposed approach.

I. Introduction

With the recent spread of portable, wireless, and hand-held computing systems, there arises more and more demands of low-power VLSI innovations[1].

The CMOS logic scheme is widely used today, in which a CMOS gate has such a structure that a pair of nMOS and pMOS logic blocks are connected serially between Vdd and Vss. Hence, while input signals are transitive between the "0" and "1" logic levels, the short-circuit current flows from Vdd to Vss, which makes the power dissipation increase suddenly according as the clock frequency grows[2]. In addition, the pass-transistor logic, which may be regarded as another promising solution[3], tends to lack the robustness against downsizing and voltage scaling[4].

The NMOS 4-phase dynamic logic[5][6] is evaluated to be still a practical low-power scheme under the current technology[7], in which four types of logic gates are used as illustrated in Fig. 1(a), together with four different clock signals as depicted in Fig. 1(b). Each type of 4-phase logic gates is composed of an nMOS logic block to realize a logic function and two nMOSFETs for precharge and gate control. Every 4-phase logic gate works in two phases, i.e. precharge and evaluation phases, and hence there exist priority relations among inputs/outputs of gates of four types as specified in Fig. 1(c). The NMOS 4-phase dynamic logic scheme has distinctive features in comparison with the CMOS logic scheme, such as elimination of short-circuit current, small capacitive load, low signal swing range, small number of transistors, and ratio-less

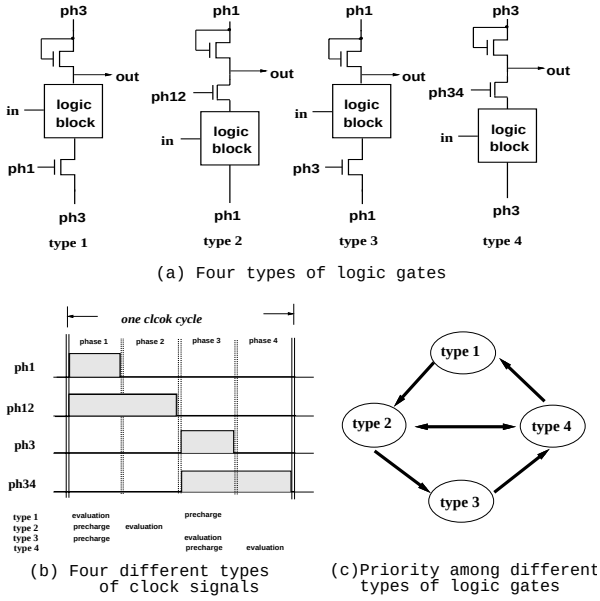

Fig. 1. NMOS 4-phase dynamic logic scheme.

transistors for layout. It is reported that not only the low-power consumption but also the short delay of a gate can be attained by this logic scheme more than that by static CMOS, domino CMOS, and pass-transistor logic schemes.

Due to this short delay performance, a logic block of this dynamic logic can contain a fairly large number of transistors, for example, 70 nMOSFETs for 100MHz, 144 nMOSFETs for 50MHz, etc[7]. Motivated by this tendency, in this paper an *Array Cell* (AC) architecture is devised, which can contain (M×N)+2 nMOSFETs to constitute an NMOS 4-phase logic gate.

II. Array Cell Architecture

A. Layout Model

For ease of layout synthesis, let a logic block be constructed in an M×N nMOSFET array as shown in

Fig. 2(a), where it should be noticed that nMOSFETs are arrayed without interconnection. Now, consider a trivial logic function which is realized by a regular interconnection as shown in Fig. 2(b), and implement this function by using 0.35μm technology at 3.3V and 1.8V supply voltages, with the nMOS threshold voltages of 0.55V and 0.4V, respectively. We have attempted SPICE simulations for this trivial function on different values of M and N. Fig. 2(c) shows a part of simulation results[7].

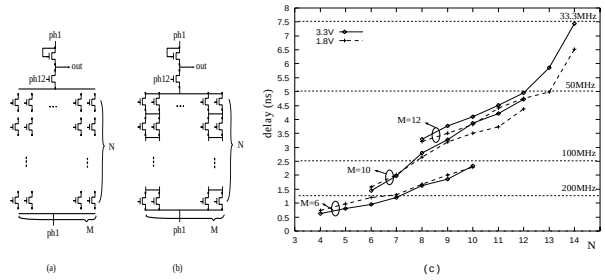

Fig. 2. Simulation results for gate delay[7].

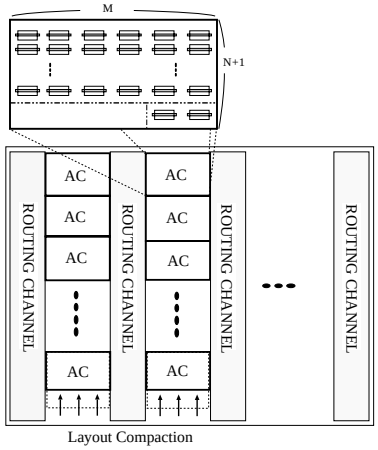

Fig. 3. Layout macro composed of array cells.

Considering that a real logic function can be constructed in this array usually with fewer nMOSFETs than this trivial function, we can see that given an operation frequency and a size of M and N, a fairly large number of logic functions can be realized in this nMOSFET array.

Now, let us define an Array Cell (AC) be a layout cell which contains M×N nMOSFETs of a logic block and two nMOSFETs for precharge and gate control, as illustrated in Fig. 3, where the number M of columns denotes the width of an AC and the number N+1 of rows the height of an AC. In terms of the layout standardization of a layout macro for the 4-phase dynamic logic, henceforth let us fix the width M of each AC, as depicted in Fig. 3, just in the same way as the standard-cell approach.

B. Layout Generation

The layout design of an AC is executed in three steps: (i)construction of an n-side graph[8] with the minimum

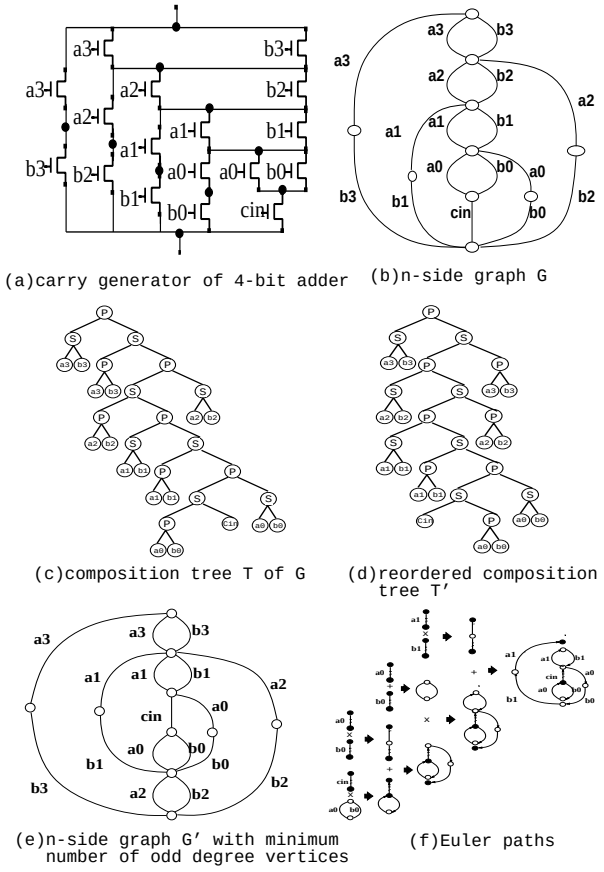

Fig. 4. Example of carry generator of 4-bit adder.

number of odd degree vertices, (ii)decomposition of an n-side graph, and (iii)placement of transistors. In what follows, the layout process is exemplified through a carry generator of 4-bit adder as illustrated in Fig. 4(a).

In realizing this logic function by means of the nMOS 4-phase dynamic logic, first an n-side graph G is constructed, which represents an interconnection topology of nMOSFETs of a logic block. Specifically, an n-side graph is constructed in such a way that

(i) each vertex u corresponds to a drain/source u of an nMOSFET, and

(ii) each edge (u,v) connecting vertices u and v corresponds to an nMOSFET whose drain/source is u and v.

Usually the interconnection topology of nMOSFETs in a logic block is of the so-called series-parallel structure, and hence, in graph theoretical terms, an n-side graph is planar[8]. Moreover, the following property of this graph is of great use for layout synthesis; if two edges i and j are adjacent in an n-side graph, then corresponding nMOSFET i and j can share a drain/source.

This series-parallel structure of G can be represented by a composition tree T[9], where T is an ordered rooted

tree in which each leaf represents an edge of G, and each interior node represents the application of a number of successive compositions of the same series or parallel operation type. Depending on the type of composition, an interior node is a *series node* (s-node) or a *parallel node* (p-node). If the ordering of the sons of a p-node is changed in T, then the graph is not changed. On the other hand, if the ordering of the sons of an s-node is changed in T, the number of odd degree vertices is reduced in the graph. Following the rule in [9], a reordered composition tree T' is constructed to achieve an n-side graph with the minimum number of odd degree vertices.

For example, an n-side graph G of the carry generator of 4-bit adder of Fig. 4(a) can be drawn as shown in Fig. 4(b). The composition tree T of Fig. 4(b) and reordered composition tree T' are drawn in Fig. 4(c) and Fig. 4(d), respectively. Fig. 4(e) shows a reordered n-side graph with the minimum number of odd degree vertices.

Given an n-side graph with the minimum number of odd degree vertices, the 2-D array layout of an AC is built. An *Integer Linear Programming*(ILP) method is used in the automatic layout generation of CMOS cells[10], where the 2-D array height minimization problem for CMOS cells is defined as a covering problem – a subset of chains that covers each transistor and has the minimum cell height, needs to be determined. However, the set of all chains which are generated from an n-side graph is too large to be solved by ILP. Considering the tradeoff between the minimum size of cell height and the computational complexity, we decompose the n-side graph into Euler paths in the following heuristic way in order to reduce the computational complexity: First, the minimum number of Euler paths are determined by tracing the reordered composition tree in the bottom up manner. Second, each of Euler paths is decomposed into the minimum number of subpaths whose maximum length is $\lceil t/M \rceil$, where t denotes the number of transistors.

Given a set of chains, the 2-D array layout is applied for the purpose of determining the placement of transistor chains. In the same manner as stated in [10], we summarize in Table I the assumptions underlying the 2-D array layout. Fig. 5 illustrates an example of the 2-D array layout for an AC. The width of AC is assumed to be fixed,

TABLE I
ASSUMPTIONS FOR LAYOUT OF AC.

1.	The logic block of a gate is composed only of series-parallel connection.
2.	AC consists of M diffusion columns.
3.	Metal-1 routing is taken as far as possible.
4.	Diffusion gaps do not allow a Metal-1 wire to be routed between them.
5.	Terminals that are on transistors placed in adjacent diffusion columns are connected using routing channel between the columns, otherwise, using routing channel on the top or bottom of cell.
6.	Fixed size of transistors are employed.

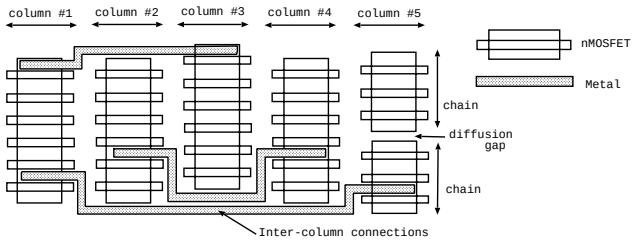

Fig. 5. Model of 2-D array layout.

TABLE II
CIRCUIT PARAMETERS OF ILP MODEL.

trans	The number of transistors
chains	The number of chains
columns	The number of columns
T, C, N	The set of transistors, chains and nets, respectively
length	Length[i] represents the number of transistors in a chain i
net	net[i,n] is 1 if chain i contains net n, and 0 otherwise

and therefore the area of AC depends only on its height. The height H_c of AC is given by the following formula[10]:

$$H_c = max\{H_j : for\ each\ column\ j = 0, \cdots, c\} \quad (1)$$

$$H_j = t_j + c_j - 1 + w_j \quad (2)$$

where c is the number of columns, and H_j, t_j, c_j, and w_j denote the height, the number of transistors, the number of chains, and the number of horizontal wires of the jth column, respectively. Table II summarizes the parameters that are input to ILP, in which each of binary variables X[i,j] is 1 if chain i is placed in column j, and is 0 otherwise.

The constraints in this ILP model are settled in what follows.

1. Chain inclusion: A chain is included exactly once.

$$\sum_{j \in 1, \cdots, c} X[i,j] = 1 \quad (3)$$

2. Column occupancy: A column must contain at least one chain.

$$\sum_{i \in C} X[i,j] \geq 1 \quad (4)$$

3. Linearizing the cost function: A non-linear cost function is made linear by expanding equation(1) into a set of linear inequalities, one for each column j.

$$H_c \geq \sum_{i \in C} X[i,j] \times length[i] + \sum_{i \in C} X[i,j] - 1 + \sum_{n \in N} h\_wire[j,n] \quad (5)$$

TABLE III
EXPERIMENTAL RESULT FOR CARRY GENERATOR.

	num.of.columns	num.of.trans.	Hc
carry generator	6	17	4
	4	17	5
	2	17	9
	1	17	17

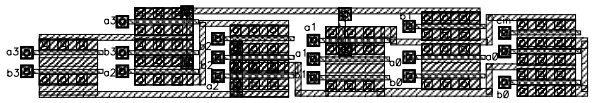

Fig. 6. Layout patterns of carry generator of 4-bit adder with 6 columns

4. Inter-column connectivity: The contribution of inter-column connections to the height of column j is measured by the number of horizontal wires that must be routed adjacent to column j. This, in turn, depends on the number of nets that connect terminals of transistors placed in columns of the left-hand side of column j to columns of the right-hand side of column j.

$$h\_wire[j,n] = left[j,n] \ \textbf{and} \ right[j,n] + \textbf{and not}\,(net[i,n] \times X[i,j]) \quad (6)$$

$$left[j,n] = \{\textbf{or}\,(net[i,n] \times X[i,j]) : columny < columnj\} \quad (7)$$

$$right[j,n] = \{\textbf{or}\,(net[i,n] \times X[i,j]) : columny > columnj\} \quad (8)$$

The above constraints are non-linear, since they involve logic operations **and**, **or**, and **not**, and hence they have to be linearized, in the same manner as stated in [10], to be included in this ILP model.

III. Experimental Results

The ILP model has been solved by using GAMS/OSL for a number of circuit examples. Table III shows experimental results for a carry generator of 4-bit adder with variable width. The layout patterns of the carry generator with 6 columns have been implemented with the area of $48.05 \times 9.55\ \mu m^2$ as depicted in Fig. 6.

For another example of a complicated logic function realized by a number of ACs, we have implemented an 8-bit round shift circuit by means of the 4-phase dynamic logic and the CMOS logic, and experimental results are summarized in Table IV. It can be readily verified that the use of our AC architecture for the 4-phase logic can reduce effectively not only the number of transistors but also the power dissipation, as compared with that of the CMOS logic.

TABLE IV
EXPERIMENTAL RESULTS FOR 8-BIT ROUND SHIFT CIRCUIT.

100MHz 3.3V	4-phase	CMOS	ratio
power (mW)	2.723	3.946	0.69
#trs.	305	476	0.64
area (μm^2)	16,418	21,561	0.76

IV. Conclusion

This paper has described an Array Cell (AC) approach and its application to the layout design. This AC architecture can integrate a given logic function with fewer transistors than the CMOS logic, resulting in low-power consumption. The layout generation of an AC is based on the n-side graph and ILP.

Development is continuing further on CAD algorithms for logic synthesis dedicated to this NMOS 4-phase dynamic logic.

References

[1] R. Rogenmoser, H. Kaeslin, and N. Felber: "The impact of transistor sizing on power efficiency in submicron CMOS circuits," in *Proc. 22nd European Solid-State Circuits Conf.*, pp. 124–127, Sep. 1996.

[2] H. J. M. Veendrick: "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE J. Solid-State Circuits*, vol. SC-19, no. 4, pp. 468–473, Aug. 1984.

[3] K. Yano, Y. Sasaki, K. Rikino, and K. Seki: "Top-down pass-transistor logic design," *ibid*, vol. 31, no. 6, pp. 792–803, June 1996.

[4] R. Zimmermann and W. Fichtner: "Low-power logic styles: CMOS versus pass-transistor logic," *ibid*, vol. 32, no. 7, pp. 1079–1090, July 1997.

[5] Y. T. Yen: "A mathematical model characterizing four-phase MOS circuits for logic simulation," *IEEE Trans. Computers*, vol. c-17, no. 9, pp. 822-826, Sep. 1968.

[6] S. P. Asija: "Four-phase logic is practical," *Electronic Design*, pp. 160-163, Dec. 1977.

[7] B.Y. Song, M. Furuie, Y. Yoshida, T. Onoye, and I. Shirakawa: "Low-power scheme of NMOS 4-phase dynamic logic," *IEICE Trans. Electron.*, vol. E82-C, no. 9, pp. 1772–1776, Sep. 1999.

[8] T. Uehara and W. M. vanCleemput: "Optimal layout of CMOS functional arrays," *IEEE Trans. Computers*, vol. c-30, no. 5, pp. 305-312, May 1981.

[9] T. Lengauer and R. Muller: "Linear algorithms for optimizing the layout of dynamic CMOS cells," *IEEE Trans. Circuits and Systerms*, pp. 279–285, Mar. 1988.

[10] A. Gupta and J. P. Hayes : "Width minimization of two-dimensional CMOS cells using integer programming," *Proc. International Conference on Computer Aided Design*, pp. 660–667, Nov. 1996.