

一种与工艺有关的 PLA 折叠法

宋俊德 辛德禄

(北京邮电学院)

1986年8月12日收到

本文提出一种新的 PLA 折叠方法叫做互补对串联折叠法 (Complementary Pair Series Folding—CPSF)。该种方法特别适用那些输入项多为互补对 (X 和 \bar{X})，而输出项并不甚多的 PLA。在一个互补对串联折叠的 PLA 中，如果每个互补对(列)都能被折叠，它的“与平面”面积可减少 40% 左右。用该种方法折叠的 PLA 的开关速度比未折叠的要高。

我们首先把 CPSF 法用于 NMOS 工艺的 PLA 折叠，设计了一个 NMOS 功率反相门作为 X 和 \bar{X} 之间的驱动器，实现了 CPSF·PLA。把 CPSF 法用于 MD-MOS 和 IIL 工艺同样获得了高速度、高装置密度的 PLA。说明利用工艺特点实现 PLA 折叠是克服 PLA 稀疏性的主要途径之一。

本文还给出了 CPSF 的算法，并讨论了与它相关的图论问题。

一、引言

PLA 已广泛地应用于大规模和超大规模集成电路 (LSI/VLSI) 的设计中。它的主要优点是版图规律性强和设计周期短。大多数 PLA 是非常稀疏的。为了改善这一缺点近年来提出了不少 PLA 折叠方法^[1-3]。也有一些其它改善稀疏 PLA 的方法^[4]。

众所周知，大多数 PLA 输入项是 X 和 \bar{X} (如 $A, \bar{A}; B, \bar{B}; C, \bar{C}; \dots$)，它们是互补的。在有些 PLA 折叠中，例如在 SCF 法中它的输入项位置受到折叠的限制，用它折叠的 PLA 输入项有时不易直接的与其它电路或外接引线相连接。文献 [1] 给出的主从 PLA 折叠法设计了一个复杂的结构较好的解决了输入端可以从任何方向引入的问题。但是这个复杂的结构本身又占去了一部分管芯面积。本文提出的 CPSF 法将不需要额外的电路结构而可解决输入项位置不受限制地引入到阵列内部。它也能较为自由地与其它电路或外接引线相连接。

一个互补对 X, \bar{X} 各自具有的积项是交错的，也就是 X 和 \bar{X} 各自的积项不可能在同一积项线上(因为 $X \cdot \bar{X} = 0$)。因此可以把任何一个互补对放在一个物理列上，这就是 CPSF 法的基本出发点。在一个 PLA 中如果每个列均可用 CPSF 法折叠，它的“与平面”面积可节省 40% 左右。这种方法特别适用于那些输入项多为互补对 (X 和 \bar{X})，而输出项较少的 PLA 中。如果设计前对 PLA 的输入端位置无任何限制而输出项数又多，则可采用该种折叠与其它折叠混合使用的方法，可节省更多的管芯面积。

一个特定形状的反向驱动器放在 X 和 \bar{X} 之间，并用 SPICE 进行了模拟。它不增加 PLA 的面积，只是把原放在 PLA 外围的驱动器移到 PLA 阵列的内部。用 CPSF 法折

叠的 PLA 与未折叠的相比，它的工作速度较高，占用管芯面积也小，并且利于计算机辅助设计 (CAD)。

本文从图论角度提出和讨论了互补对串联折叠的算法。输出低电平和多晶硅电阻之间的关系用 SPICE 进行了模拟。并设计了一个包含几个折叠列 (CPSF 法、SCF 法) 和未折叠列组成的测试管芯，对它们的电气及其它性能作了比较。

二、物理设计

缩短 VLSI 管芯设计周期的关键是增加管芯版图的规律性。PLA 是实现版图规律化有利于采用 CAD 的方法之一。它简化了 LSI/VLSI 的设计，缩短了设计周期。PLA 设计的几个过程：功能设计、版图设计和物理设计都围绕着两个目的：缩小所占用的管芯面积和提高工作速度。

功能设计：利用布尔代数化简规则以求得最少输入项数和最少积项数。该点是在作 PLA 布局之前的重要化简步骤之一。它将直接影响 PLA 占用面积。这些化简方法和理论已比较成熟，近年来也发表了不少这方面的文章。

版图设计：为了克服 PLA 的稀疏性，人们想出了各式各样的方法，如直接压缩掉 PLA 无积项的部分；采用内部路途法解决 PLA 稀疏性的方法等^[6]。但是近年来取得比较系统和全面进展的要算各式各样的折叠 PLA 的方法。例如文献[1]用了总线结构去改善折叠的 PLA 功能和性能；文献[5]用了贮存/逻辑阵列设计了一个折叠的 PLA 的版图，文献[2]、[3]和[4]给出了 PLA 简单列折叠 (SCF)、简单行折叠 (SRF)，简单行列折叠 (SCRF) 和混合的行列折叠 (MCRF) 等方法。并给出了各自的算法和实现过程。所有这些折叠方法都是为了减少占用管芯面积和提高工作速度。

物理设计：包括利用现有的各种工艺结构特点寻求版图和工艺结合以求得更好的折叠方法。本文提出的互补对串联折叠法就是利用了 NMOS 标准工艺结构巧妙的设计了一个反向驱动器而较好地解决了 PLA 的稀疏性和工作速度的问题。它也为采用 CAD 提供了可能和方便，其算法也得到了简化。

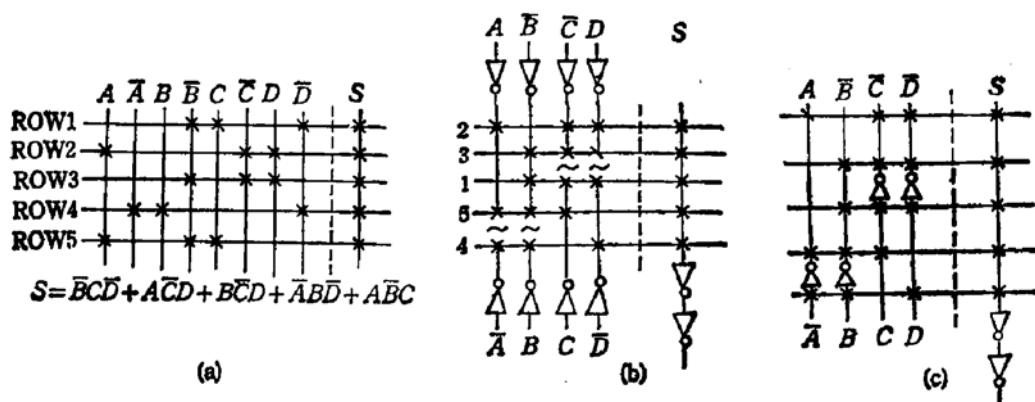


图 1

为了说明 PLA 的折叠问题，图 1(a) 给出了一个 PLA 的内部阵列示意图。图中

$A, \bar{A}; B, \bar{B}; C, \bar{C}; D, \bar{D}$ 是输入项, S 是输出项。图 1(b)是用 SCF 法对图 1(a)的 PLA 作了折叠。可见 A 和 \bar{A} , B 和 \bar{B} , C 和 \bar{C} , D 和 \bar{D} 分别安排在 PLA 阵列两边: A, \bar{B}, \bar{C}, D 是来自 PLA 的顶部, \bar{A}, B, C, \bar{D} 来自 PLA 的底部。也就是说 A 和 \bar{A} (或 B 和 \bar{B} , C 和 \bar{C} , D 和 \bar{D}) 是从两个方向引入到 PLA 阵列的内部。

如果设计前我们对集成电路输入端位置 (假如它们正是 PLA 的输入) 有一定的要求, 或者 PLA 的输入线要与其它电路相连接, 而这些电路位置又有一定的约束。在上述情况下, SCF 法必须使用另外一些金属线或者其它一些结构以解决 PLA 因折叠后位置受到约束的问题。文献 [1] 有一个非常复杂的总线结构解决了这个问题。但这个结构本

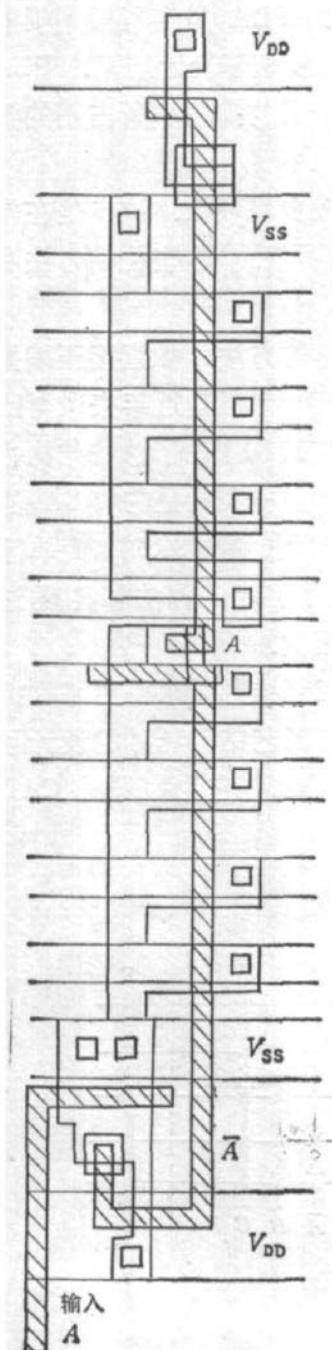


图 2(a)

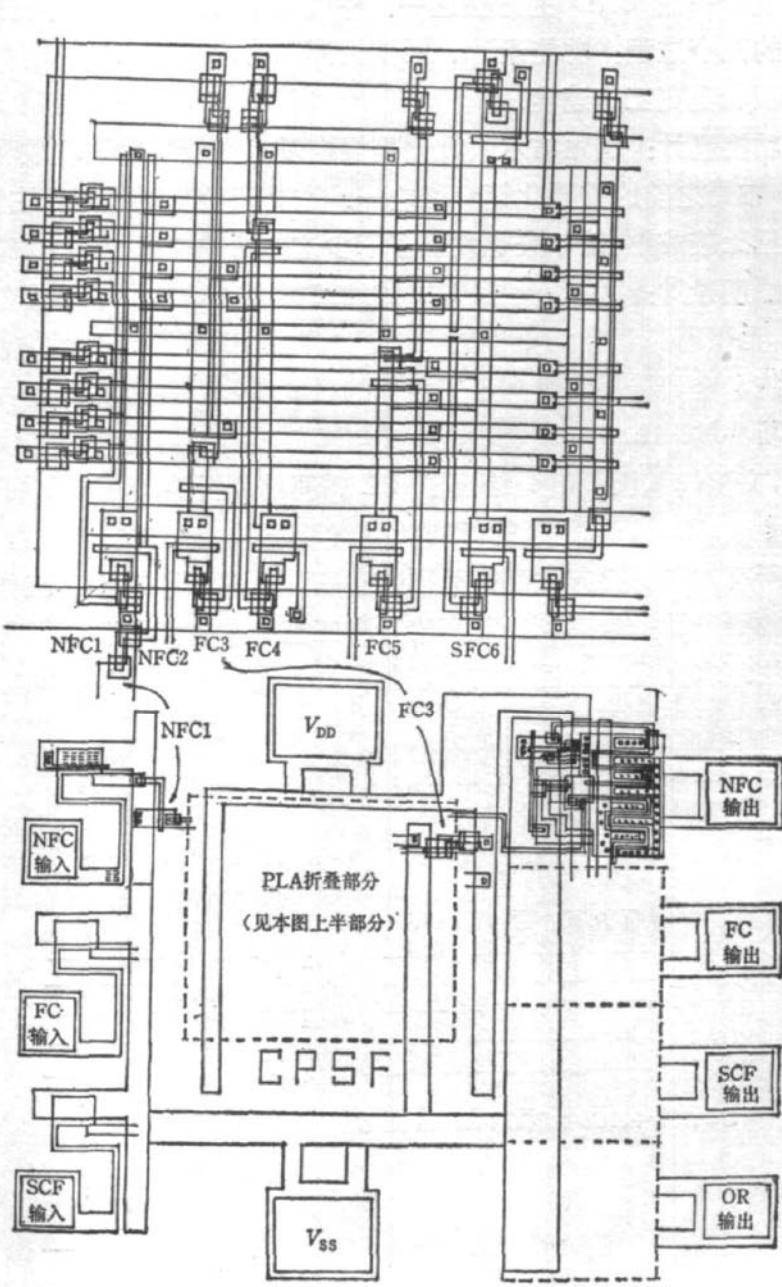


图 2(b)

身又占去了一些管芯面积。因此我们应该估价由于折叠而省下的面积是否比这个复杂结构占去的面积更多。否则，在该种条件下不能采用 SCF 法。

把一个 MOS 反相驱动器串联在 X 和 \bar{X} 之间 (A 和 \bar{A} , B 和 \bar{B} , C 和 \bar{C} 之间等……)，可获得一个互补串联折叠对如图 1(c) 所示。显然这个驱动器被安排在阵列内部。我们需要精心设计，且遵循现有设计规则并使它不改变原 PLA 的尺寸关系(如：列与列间距和行与行间距等)。这就需要利用一切可利用的面积设计一个反相驱动器且实现性能最佳。驱动器的版图可以有多种，图 2(a) 给出了一个设计实例。用这种方法折叠的 PLA，它的输入端位置可以相当自由。因为它的 X 和 \bar{X} 两个引入端串联后成为同一个列。所以从 X 到 \bar{X} 或从 \bar{X} 到 X 均可，只要把反相器的版图倒相 180° 即可。反相驱动器可以放在阵列中任何一行的位置上。输入端可以从上或从下部引入。也可以根据管壳引线或其它电路连线位置而定。这为设计带来了极大方便。图 2(a) 的反相驱动器中耗尽型和增强型晶体管版图相关比率系数为^[8]：

$$\text{比率系数} = \frac{\text{耗尽型管栅长度 / 耗尽型管栅宽度}}{\text{增强型管栅长度 / 增强型管栅宽度}} = 4-5.$$

为了对 CPSF 法、SCF 法及未折叠的列进行电气性能模拟、比较和测试。一个完整的包括上述几种不同列的测试管芯图示于图 2(b)。此图的结构、布局和相关尺寸采取了类似[8]和[9]的设计，以利比较。用 CPSF 法折叠的几个列反相驱动器被放在各列的不同位置上。以此说明该种设计实现任何位置折叠的可能性。图 2(b) 的上图中，从左起第一、二列 NFC1、NFC2 是未折叠的列。第三到第五列 FC3、FC4、FC5 是用 CPSF 法折叠的列，说明驱动器的位置在不同部位都可以实现。第六列是用 SCF 法折叠的列，它有两个输入：一个在顶部，一个在底部。

首先我们把 NFC1、NFC2 与 FC5 的电特性作一比较。它们的逻辑功能相同但版图不同。它们的等效电路分别如图 3(a) 和 3(b) 所示。R 表示多晶硅寄生电阻。

$$R = R_{\square} \times \frac{L}{W}.$$

R_{\square} : 多晶硅方块电阻, L : 多晶硅条的长度, W : 多晶硅条的宽度, C 表示寄生电容，它包括多晶硅到基片的电容、金属层到多晶硅的电容以及结面积电容。

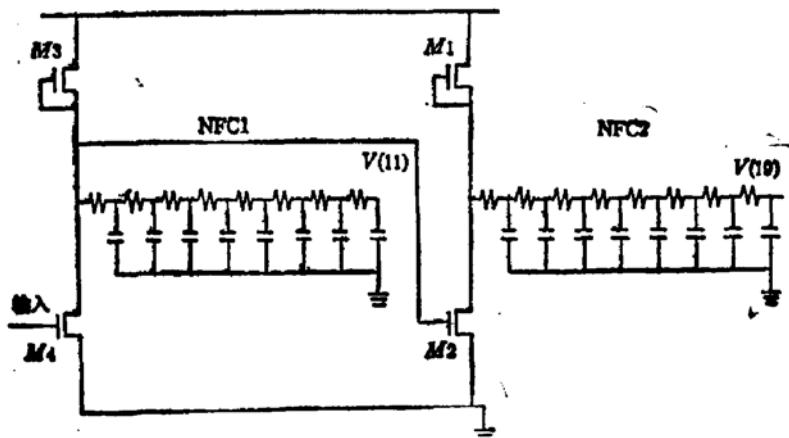


图 3(a)

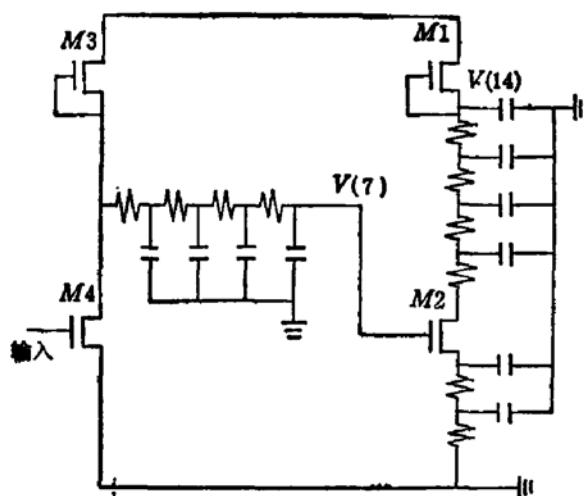


图 3(b)

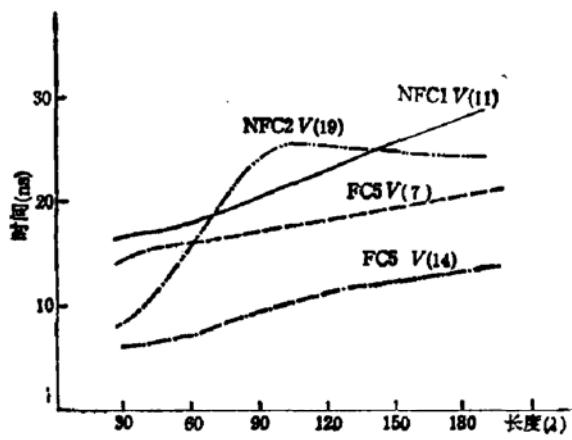


图 3(c)

为了利于作性能比较,在作 SPICE 模拟时,我们采取的参数数值均参考 [8]、[9] 以及美国 GI 公司制备时提供的实际运用值。在图 3(a) 和 3(b) 中我们选用了 NFC1 的点 $V(11)$ 和 NFC2 的点 $V(19)$ 分别与 FC5 的点 $V(7)$ 和点 $V(14)$ 相比较。它们在逻辑功能上是一一对应的点。它们的延迟时间用 SPICE 进行了模拟。结果如图 3(c) 所示。输入输出波形表示在图 3(d) 和 3(e) 中。由图 3(c) 可见用 CPSF 折叠后的点 $V(14)$ 和 $V(7)$ 的速度分别比点 $V(19)$ 和 $V(11)$ 要快。在 NFC 中 $V(19)$ 的曲线有一个峰值,可能是由于寄生电容和电阻引起的。尽管如此,它的延迟时间仍要比对应点 $V(14)$ 长。

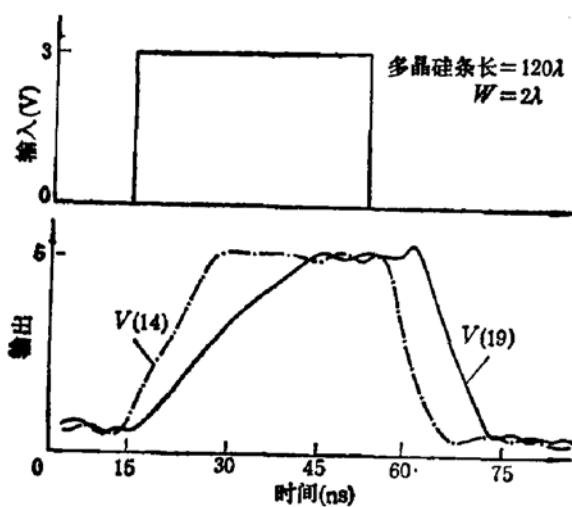


图 3(d)

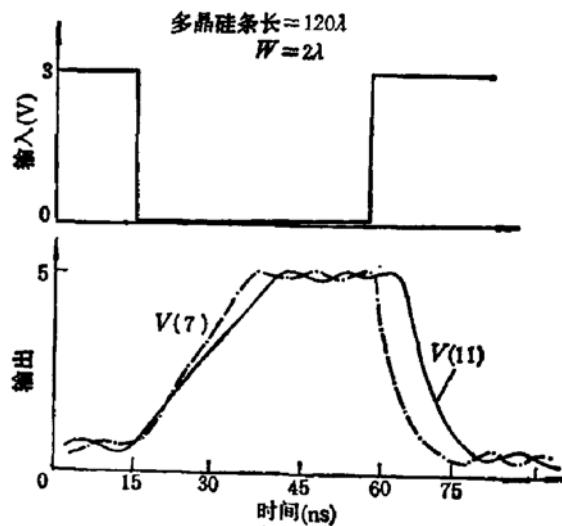


图 3(e)

在互补对串联折叠的 PLA 中,关键点是电流流过多晶硅条。在图 3(b)中当点 $V(7)$ 电压为高电平时,电流将流过 M_1 、 M_2 和多晶硅条。所以多晶硅的电阻值将直接影响输出电压 $V(14)$,对输出电压低电平值的影响取决于 M_1 的等效电阻和多晶硅电阻的比值(图中 M_3 导通时的等效电阻可以忽略)。

我们用 SPICE 也模拟了输出电压低电平与多晶硅等效长度的关系。它的结果如图

4 所示。多晶硅方块电阻取 $35-45\Omega/\square$ 。例如，当这个多晶硅宽度取 2λ ，长度为 340λ 时，输出电压低电平将低于 $0.45V$ 。可见在互补对串联折叠 PLA 中，安排 80—85 个积项是可行的。

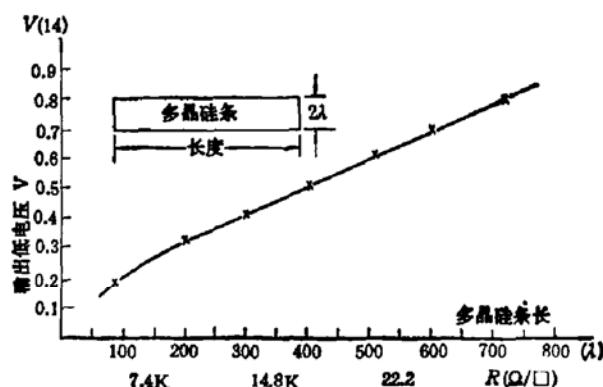


图 4

三、互补对串联折叠 PLA 的算法

在我们讨论互补对串联折叠的算法之前，让我们先介绍下列诸定义：

1) 互补折叠对 CFP (Complementary Folding Pair)，它是一个折叠对

$$f_i = \{C_i, C_{i+1}\}.$$

式中 C_i 和 C_{i+1} 代表 PLA 中两个互补的列 (X 和 \bar{X})，如 $C_i = A$ ，则 $C_{i+1} = \bar{A}$ (当 $i = 1$)； $C_i = B$ ，则 $C_{i+1} = \bar{B}$ (当 $i = 3$)；这里 i 可以是 $1, 3, 5, \dots$ 。例如图 5(a) 中 A 和 \bar{A} 是一个互补折叠对。

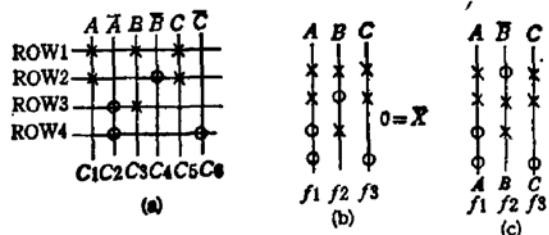


图 5

2) 顺序互补折叠对 OCFP (Ordered Complementary Folding Pair)。它的定义如下：

$$O_i = (C_i, C_{i+1}).$$

式中 C_i 含有所有积项必须处于 C_{i+1} 所含有各积项的上面(或下面)。我们注意到每一对 C_i 和 C_{i+1} 首先必须是一个互补折叠对，然后才能判断是否可构成顺序互补折叠对。例如图 5(b) 中 A 和 \bar{A} 是一个 OCFP； B 和 \bar{B} 则不是一个 OCFP。因为 B 所有各积项并非全在 \bar{B} 的上部(或全在 \bar{B} 的下部)。

3) 互补折叠集 CFS (Complementary Folding Set)：

$$F = \{f_1, f_2, f_3, \dots, f_k\}$$

是一个所有互补折叠对的集。式中 f_1, f_2, \dots, f_k 是互补折叠对。例如图 5(c) 给出一个 CFS。

4) 顺序互补折叠集 OCFS (Ordered Complementary Folding Set)：

$$O = \{O_1, O_2, \dots, O_k\}$$

是所有顺序互补折叠对的集。式中 $O_1, O_3, O_5, \dots, O_k$ 是顺序互补折叠对。图 5(c)同时也是一个 OCFS。

本文将从图论角度出发首先引出列交连图，顺而解释和讨论互补对串联折叠 PLA 问题，并从此导出 CPSF 的算法。

一个无向图 $G = G(V, E)$ 代表一个列交连图，式中 V 是点的集， E 是无向边的集它们的关系可描述如下：

$$\text{集 } E = \{e = \{v_i, v_j\} / R(C_i) \cap R(C_j) \neq \emptyset\}$$

式中 v_i 和 v_j 表示用无向边 e 连接起来的两个点。实质上它们代表 PLA 阵列中有共同

连接边的两个列。 i 和 j 代表点集中的不同的两个点的顺序数。所以 $R(C_i)$ 与 $R(C_j)$ 的交集不是空集。图 5(a)中的 PLA 的交连图划在图 6(a)。

我们把 OCFS 和它们交连边集划在同一个图中，得到一个混合图 $G(V, E, \hat{E})$ ，如图 6(b) 所示，式中 \hat{E} 表示方向边。

图 6

$$\hat{E} = \{\hat{e}_{i,i+1} = (v_i, v_{i+1}) / R(C_i) \cap R(C_{i+1}) = \emptyset\}.$$

v_i 和 v_{i+1} 是由方向边连接起来的两个点，实质上它们代表 PLA 阵列中无共同连接边的一个互补对。所以列 C_i 和 C_{i+1} 的交集是空集(因为 $X \cdot \bar{X} = 0$)。

在互补对串联折叠的 PLA 中选择 θ 的方向是十分重要的。因为表示互补的两个列关系的每个 $\theta(A_1, A_2, A_3, \dots)$ 存在着两种可能的方向。取二者之一说明 X 或 \bar{X} 哪一个将在折叠后的 PLA 的顶部。如图 6(b) 中方向边 A_1 可有两个方向： $C_1 \rightarrow C_2$ 或者 $C_2 \rightarrow C_1$ ；如果我们选前者 $C_1(A) \rightarrow C_2(\bar{A})$ ，即说明 C_1 在顶和 C_2 在底。若取 $C_2 \rightarrow C_1$ 则与上相反。在图 6(b) 中我们若同时取 $C_1(A) \rightarrow C_2(\bar{A})$ 和 $C_3(B) \rightarrow C_4(\bar{B})$ 的方向，即 A_1 是 $C_1 \rightarrow C_2$ 和 A_2 是 $C_3 \rightarrow C_4$ ，这将导致出现一个“Cycle”(或称交连环)*，即它们的连接按照下述顺序而构成了一个环， $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_1$ (此处“ \rightarrow ”表示一个方向边；“—”表示一个无方向边)这是因为在顺序折叠集中。

$$O = \{(C_1, C_2)(C_3, C_4)\}$$

其中包括下面的关系(见图 6(a) 和 6(b))

$$Q(O) = \{r_1 < r_3, r_1 < r_4, r_2 < r_3, r_2 < r_4, r_1 < r_2, r_3 < r_2\}.$$

式中 r_1, r_2, r_3 和 r_4 表示某列在行 1、行 2、行 3 和行 4 分别有一个“控制”(Care)，而 $r_1 < r_4$ 说明第一列的 r_1 在第二列的 r_4 的上面，由上式可见， $r_2 < r_3$ 和 $r_3 < r_2$ 是矛盾的，因此可以判定它们是不可实现的，从图论角度看在上述情况下用 CPSF 法会产生一个“Cycle”。所以在这种情况下， A_1 和 A_2 不可能同时用 CPSF 法。如果我们改变 A_2 的方向为 $C_4(\bar{B}) \rightarrow C_3(B)$ ，这时 C_1, C_2, C_3 和 C_4 间将不存在一个“Cycle”，这是因为：

$$Q(O) = \{r_1 < r_3, r_1 < r_4, r_2 < r_4, r_2 < r_3, r_2 < r_1, r_3 < r_1\}.$$

上式不存在任何矛盾，它是可安排的。从图论角度解释即它们不再存在一个“Cycle”(见

* “Cycle” 定义见参考文献 [2]。

图 6(b)). 可见在 CPSF 中运用好这一点在同样条件下可以实现更多列折叠，在下述的 CPSF 算法中包括了这一点。根据上述定义和讨论给出互补对串联折叠算法的步骤如下：

第一步，对输入项中所有的互补折叠对 CFP 冠以点顺序数，如

$$A_1 \rightarrow v_1, \bar{A} \rightarrow v_2, B \rightarrow v_3, \bar{B} \rightarrow v_4 \dots$$

$$X \rightarrow v_i, \bar{X} \rightarrow v_{i+1} \quad (i = 1, 2, 3, \dots)$$

第二步，按照顺序数逐步对每个 CFP 加入后的交连图进行测试，以得知哪些列是可安排于同一个互补对串联折叠的 PLA 中。例如，首行取第一个 CFP，即 v_1 和 v_2 总是可安排的，因它不受任何约束。所以继续作的是接受它，并记入“增长集”(Growing Set) 内。

第三步，取下一个 CFP，并首先确定某一方向，例如 $v_i \rightarrow v_{i+1}$ ；测试加入这个 CFP 后是否产生了“Cycle”。如果没有，则继续按顺序取下一个 CFP 并重复上述步骤。直到发现由于一个新的 CFP 的加入而产生了“Cycle”：1) 改变 $v_i \rightarrow v_{i+1}$ ；测试是否“Cycle”继续存在；2) 如果继续存在“Cycle”则要删除这一 CFP；如果不存在“Cycle”则接受这一 CFP 并记入增长集。

第四步，重复第三步直到取完所有 CFP 为止。

CPSF 的算法如下：

DATA: $G = G(V, E)$

INITIALIZATION: $\hat{E} \leftarrow \emptyset$

$\theta_i\{A(V_1), B(V_2), C(V_3)\dots\}$

$\theta_{i+1}\{\bar{A}(V_2), \bar{B}(V_3), \bar{C}(V_4)\dots\}$

MAIN LOOP: WHILE (SET θ_i IS NOT EMPTY)

10: TAKE A VERTEX v_i FROM SET θ_i BY ORDER ($i = i$)

WHILE (SET θ_{i+1} IS NOT EMPTY)

TAKE A v_{i+1} FROM θ_{i+1} TRY $v_i \rightarrow v_{i+1}$

IF {A CYCLE IS CREATED BY THE EDGE FROM v_i TO v_{i+1} IN THE MIXED GRAPH $G(V, E, \hat{E})$ }

THEN $v_{i+1} \rightarrow v_i$

IF {THERE IS A CYCLE AFTER THE CHANGE}

THEN REMOVE v_{i+1} AND v_i

ELSE {APPEND TO \hat{E} WITH $(v_{i+1} \rightarrow v_i)$;

UPDATE: REMOVE v_{i+1} FROM θ_{i+1} ;

REMOVE v_i FROM θ_i ;

IF $\theta_i = \emptyset$ OR $\theta_{i+1} = \emptyset$

THEN END

ELSE GOTO 10 }}

END

参 考 文 献

- [1] Roy A. Wood, *IEEE Transaction on Computer*, C-18, 602 (1979).
- [2] G. D. Hachtel, A. R. Newton, and A. L. Sangiovanni-Vincentelli, *IEEE Transaction on CAD of Integrated Circuits and System*, CAD-1, 63 (1982).
- [3] M. W. Stebnisky, M. J. Meginnis J. C. Werbichas, R. N. Putatunda and A. Feller, Proc. of the 1982 International Conference on Circuits and Computers, 156, (1982).
- [4] G. D. Hachtel, A. R. Newton, and A. L. Sangiovanni-Vincentelli, 19th Design Automation Conference, 147 (1982).
- [5] G. D. Hachtel, A. L. Sangiovanni-Vincentelli, A. R. Newton, 18th Design Automation Conference, 1023 (1981).
- [6] Samuel Chuquilanqui, 20th Design Automation Conference, 795 (1983).
- [7] Z. Arevalo, J. G. Bredeson, *IEEE Transaction on Computer* C-27, 1028 (1978).
- [8] C. A. Mead and Conway, *Introduction to VLSI System*, Addison-Wesley Publishing company (1980).
- [9] Robert W. Hon and Carlo H. Sequin, *A guide to LSI Implementation*, Xerox Palo Alto Research Center, 140 (1980).

A PLA Folding Method Related to Processes

Song Junde and Xin Delu

(Beijing University of Posts and Telecommunications)

Abstract

A new PLA folding method, Complementary Pair Series Folding (CPSF), is presented. It is particularly suitable to the use of PLA, in which most of the inputs are complementary pairs (X and \bar{X}), besides, there are not too many outputs. If each input column is folded, the "AND PLANE" area of PLA with CPSF can be reduced by as much as forty percent. The switching speed of the CPSF structure is faster than unfolded structure.

PLA with CPSF is accomplished by using NMOS process. A NMOS inverter is designed as a driver between X and \bar{X} . Higher packing density and faster switching speed are attained as the PLA with CPSF applied to the MD-MOS and IIL processes. It is an important method to improve the sparseness of PLA by the use of some process peculiarities.

An algorithm of CPSF is also given, and the related graph theory is discussed.