D. T. Lee
S. J. Hong
C. K. Wong

Number of Vias: A Control Parameter for Global Wiring of High-Density Chips

In integrated circuits, components are frequently interconnected by horizontal and vertical wires in respective wiring planes whether on chip, card, or board. The wire changes direction through "vias" that connect the orthogonal wiring planes. Because of technology constraints, the arrangement of vias must conform with certain neighborhood restrictions. We present results on the guaranteed minimum number and maximum possible number of vias in a given wiring cell for various technology constraints. These numbers provide an early means of control on global wiring routes to further the success of the exact embedding process that follows global wiring.

1. Introduction

The masterslice or the gate array is a common arrangement of components and wire tracks on LSI or VLSI chips [1-5]. The logic gates are placed in a rectangular array, and the terminals of these gates are interconnected by horizontal and vertical wire tracks on the respective wiring planes. The group of available wiring tracks in a row or a column of gates is called the horizontal/vertical wiring channel. The wires change directions by accessing the orthogonal wiring plane through "vias" programmed at the intersections of horizontal and vertical wire tracks. The intersection of the horizontal and vertical channels, together with the corresponding gate component, is called a *cell*.

Wiring of a masterslice chip usually proceeds in two phases [3, 4]. The first phase is the global routing of nets, *i.e.*, the connections between terminals that have to be electrically common. The route of each net wire is determined in cell resolution. The exact embedding phase that follows then determines the detailed assignment of wire segments globally allocated to the cells to the individual tracks.

The first objective of global wire routing is usually to avoid violating the channel capacity of any cell boundary (uniform wiring density, if possible); the second objective is to produce a near-minimal total wire length. In deciding upon a particular connection segment, the global wiring process finds several (or all if exhaustive) feasible routes and selects a near-minimal-length path that does not "crowd" the cell boundaries en route. It usually involves a cost function reflecting the number of tracks available at each cell boundary. The path cost is then the sum of all the boundaries crossed by the path. Of course, some global algorithms are greedy and choose the first (usually the shortest) route found to be feasible, with the consequence that the success rate of the global wiring process may decrease.

2. The via problem and summary of results

The wire routing problem arises mainly because there are a limited number of tracks available. As the number of components and the proportional number of nets grow in VLSI, the difficulty is compounded and more is demanded of the global process to ensure the success of overall wiring. Even when all the nets are routed successfully at the end of the global phase, some of the wire segments may not be embeddable in the tracks within the global route. The resulting overflow wires must be handled by time-consuming manual re-routing and embedding. One of the main causes of this problem is the via placement restriction.

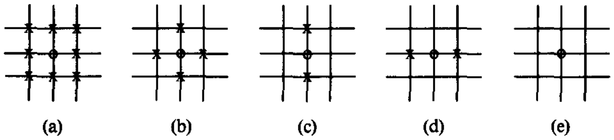


Figure 1 Via restrictions: (a) 8-neighbor exclusion; (b) 4-neighbor exclusion; (c) vertical exclusion; (d) horizontal exclusion; (e) no exclusion.

Because of various technology constraints vias cannot be placed in certain configurations. Some of the most prevalent restrictions are shown in Fig. 1. For a given via position (o), the forbidden neighbor positions (x) for other vias are shown on the track intersections.

What happens if the vias are not considered at the global phase can be illustrated by a simple example. Consider a cell with one wiring track in the horizontal and one in the vertical direction. The global routing may assign two bending wires to the cell without violating the channel capacities. But there is only one via placement possible at the cell even if there were no via exclusions! In general, for a given cell of n horizontal and m vertical tracks (referred to as an $n \times m$ cell), there is a maximum number of vias that can be allowed for each via placement restriction. Beyond this maximum number, $V(n, m)$, no clever arrangement of vias can satisfy the restrictions. Obviously, $V(1, 1) = 1$ for all restrictions.

On the other hand, there is a minimum number, $v(n, m)$, of vias that can be placed in a cell such that if any number less than $v(n, m)$ is embedded in the cell, one can always find another legal via position, regardless of the configuration of vias. Therefore, if the vias allocated in a cell do not exceed $v(n, m)$, it is almost certain that the exact embedding will not fail in the cell due to the via placement restrictions.

Given these two parameters, for each cell on the chip, most global routing algorithms can be easily modified to accommodate the via restrictions. That is, when the selection is to be made from many feasible routes for a segment, the cost of vias must be added to the cost of each route. The cost of each via en route can be computed as follows.

1. It is almost zero if the number of vias assigned in the cell so far is less than $v(n, m)$;
2. It increases exponentially if the number of vias already used in the cell is between $v(n, m)$ and $V(n, m)$, such that the cost is infinite if the number used equals $V(n, m)$.

The infinite cost means that the route is not a feasible one, *i.e.*, no additional bending wires can be globally allocated to the cell. A via cost of this nature can be adopted for any specific global routing algorithm.

The calculation of the minimum and maximum number of vias is somewhat involved. We first derive the $v(n, m)$ and $V(n, m)$ values for the eight-neighbor exclusion case and then treat the other cases in a similar fashion. For ease of explanation we cast the problem in terms of a game on an $n \times m$ chess board with pebbles representing where the vias are placed. Once a pebble, x, is placed at position (i, j), none of its eight neighboring cells (not the cells of the chip, which is the whole chess board in this game) may contain another pebble. We say these excluded cells are covered by *distance* to x. Furthermore, each placed pebble excludes another pebble being placed in any of the four L-shaped lines bending at the pebble and extending to the two orthogonal boundaries of the chess board. We say these cells are covered by the *line* of the pebble x. Figure 2 illustrates the four possible coverings by line.

The values of $v(n, m)$ and $V(n, m)$, the minimum and maximum number of pebbles placeable on the board under various conditions, are summarized in the following equations. The values for small n and m are listed in Tables 1-5, shown later.

1. Neighboring condition (a)

$$v(n, n) = n - 3, \qquad \text{for } n \ge 7. \tag{1}$$

$$v(n, n + 1) = n - 2, \qquad \text{for } n \ge 5. \tag{2}$$

$$v(n, n + 2) = n - 2, \qquad \text{for } n \ge 7. \tag{3}$$

$$v(n, n + 3) = n - 1, \qquad \text{for } n \ge 3. \tag{4}$$

$$n(n, n + 4) = n - 1, \qquad \text{for } n \ge 5. \tag{5}$$

$$v(n, n + 5) = n, \qquad \text{for } n \ge 1. \tag{6}$$

$$v(n, m) = n, \qquad \text{for } m > n + 5, n \ge 1, \tag{7}$$

and

$$V(n, m) = 2 \min (n, m) \quad \text{if } \max (n, m) \ge 9. \tag{8}$$

2. Neighboring conditions (b), (c), (d), and (e)

$v(n, m) = \min (n, m)$ under conditions (b), (c), (d), and (e).

$$V(n, m) = \begin{cases} 2 \min (n, m) \text{ for } \max (n, m) \ge 4, \text{ under conditions (b), (c), and (d);} \\ 2 \min (n, m) \text{ for } \max (n, m) \ge 2, \text{ under condition (e).} \end{cases}$$

Note that condition (d) just means transposing Table 5.

Sections 3 and 4 are devoted to the establishment of values of $v(n, m)$ under condition (a). Section 5 is for the value of $V(n, m)$ under the same condition. In Section 6, other conditions are considered.

3. Lower bounds for $v(n, m)$ under condition (a)

We first have the following two obvious lemmas.

• *Lemma 1*

$v(n, m) \le V(n, m) \le 2 \min(n, m)$.

Proof

It follows from the fact that each row and each column can have at most two pebbles.

• *Lemma 2*

$v(n, m) \le v(n', m')$,

$V(n, m) \le V(n', m')$,

for $n \le n'$, and $m \le m'$.

Proof

Immediate.

In the following, we first prove the lower bounds for $v(n, m)$.

• *Lemma 3*

$$v(n, n + 1) > n - 3, \qquad n \ge 5. \tag{9}$$

Proof

Suppose the assertion is not true. Then there are at least three empty rows and four empty columns.

1. We first show that the bottom row cannot be empty. Since there are at least four empty columns, if the bottom row were empty, the four cells a, b, c, and d, *i.e.*, the intersection of the bottom row and the four empty columns, must be covered by distance. Assume that pebble x_1 covers cells a and b and x_2 covers cells c and d, as shown in Fig. 3(a). These two pebbles cover exactly six cells. The remaining $(n + 1) - 6 = n - 5$ cells must be covered by line and require at least $n - 5$ pebbles. Excluding x_1 and x_2 we have exactly $n - 5$ pebbles available. Let us denote the two zones, each three columns wide, by A_1 and A_2, as shown in Fig. 3(a). Thus, we have the following *placement conditions*, *i.e.*, we require (a) that the remaining pebbles be placed outside the zones A_1 and A_2 and (b) that each column receive exactly one pebble. It is easy to see that, if either of the placement conditions is violated, we will have a contradiction. Now, due to the placement of two pebbles x_1 and x_2 in the same row, there are at least three more empty rows. Let us consider the top two empty rows R_1 and R_2, as shown in Fig. 3(b) such that row R_1 is at least one row apart from R_0 in which x_1 and x_2 lie. Consider then the four designated cells of R_1 and R_2 in zones A_1 and A_2, respectively [Fig. 3(b)]. These designated cells must be *covered by distance*. Since each row can have at most two pebbles, to cover these eight cells while satisfying placement condition (a) we will be forced to place two pebbles in a column, violating condition (b).
2. Suppose that the bottom i rows are nonempty, $i \ge 1$, and the $(i + 1)$th row is empty. We prove that none of these i nonempty rows can be occupied by two pebbles. To show this let us assume that the jth row, $1 \le j \le i$, is occupied by two pebbles x and y and that all the rows below j are occupied by one pebble. Observe that the pebbles in rows below j *cannot* cover by distance any cells in rows above the jth row. Although the pebble in the $(j - 1)$th row can *cover by distance* some cells of the jth row, these cells, in fact, all the cells of the jth row, are covered by pebbles x and y. We can therefore remove the $j - 1$ pebbles below the jth row and the columns occupied by them without affecting the "coverability" of the remaining pebbles covering

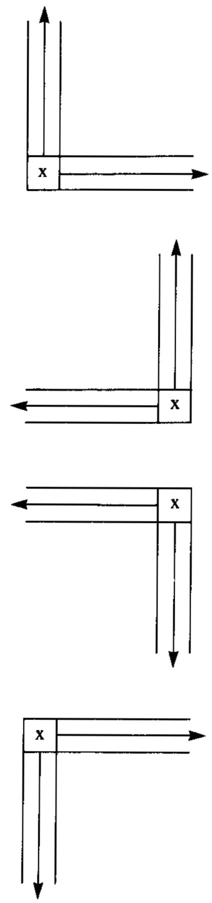


Figure 2 Four possible coverings by line of x.

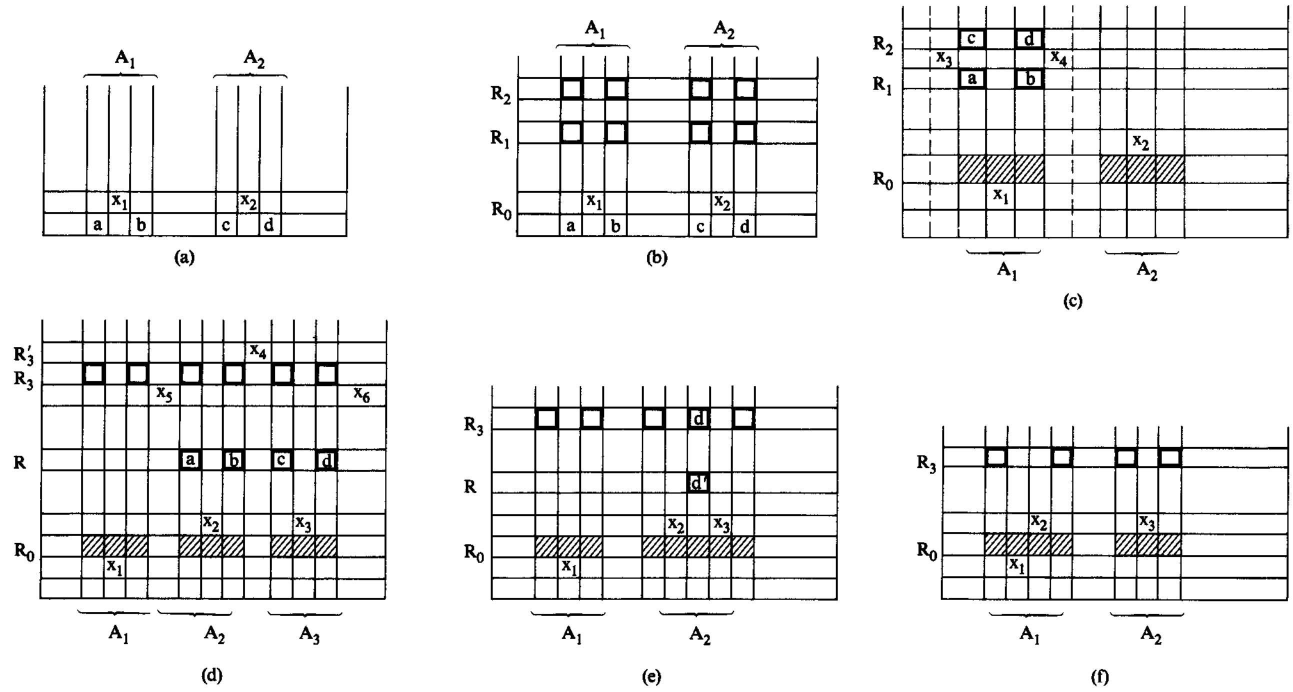


Figure 3 Illustration for the proof of Lemma 3.

the remainder of the board with $n - (j - 1)$ rows and $(n + 1) - (j - 1)$ columns. That is, if $n - 3$ pebbles are sufficient to cover a board of size n by $n + 1$, with the removal of $j - 1$ pebbles, the remaining pebbles $(n - 3) - (j - 1)$ must be able to cover the remainder of the board. But by induction, $v(n - (j - 1), (n + 1) - (j - 1)) > n - (j - 1) - 3$, *i.e.*, $n - (j - 1) - 3$ pebbles are not sufficient to cover a board of size $n - (j - 1)$ by $n - (j - 1) + 1$, and we conclude that $n - (j - 1) - 3 + (j - 1) = n - 3$ pebbles are not sufficient, a contradiction.

3. We therefore can assume that if row R_0 is the first empty row from the bottom, then there exists at least one nonempty row below it and, furthermore, these nonempty rows are each occupied by one pebble. Similar results hold for the first empty row from the top. Now let us distinguish two cases: (a) R_0 is covered by distance by two pebbles, and (b) R_0 is covered by distance by three pebbles. (Recall that the row below R_0 is occupied by only one pebble.)
   (a) R_0 is covered by distance by two pebbles, x_1 and x_2. If x_1 and x_2 cover by distance only five or fewer cells of R_0, then the remaining cells must be covered by line and require at least $n - 4$ pebbles, a contradiction. Thus, x_1 and x_2 must cover six cells of R_0. Let us denote the two zones induced by pebbles x_1 and x_2 by A_1 and A_2 [Fig. 3(c)]. These two zones may or may not be adjacent. Since the remaining $n + 1 - 6 = n - 5$ cells of R_0 are to be covered by line and we have $n - 5$ pebbles available, we have the same placement conditions as in case (1). Recall now that we have two more empty rows, R_1 and R_2, above R_0. Consider the four designated cells a, b, c, and d of R_1 and R_2 in zone A_1 [Fig. 3(c)]. In order to cover these four cells without violating the placement conditions, rows R_1 and R_2 must be one row apart, and two pebbles, x_3 and x_4, must be placed in the row between R_1 and R_2 and in the columns adjacent to zone A_1, as shown. Due to the placement of two pebbles in one row, we can find an additional empty row, R_3. To cover the two cells of R_3 in zone A_1, we will need to place pebbles in the same columns as x_3 and x_4, violating condition (b).
   (b) R_0 is covered by distance by three pebbles x_1, x_2, and x_3 ($n \geq 6$). In this case we have two pebbles in the row above R_0 and one below R_0, and we have $n - 6$ pebbles available and three more empty rows R_1, R_2, and R_3. Assume that R_3 is the topmost empty row. Note that the row R_3', which is above R_3, must be occupied by exactly one pebble by the same arguments given in (2). We distinguish several cases depending on the number of cells of R_0 that are covered by distance by x_1, x_2, and x_3. In

fact, we only need to consider four cases, *i.e.*, they cover by distance 7, 8, and 9 cells of R_0.

Case 1. Nine cells [Fig. 3(d)]. There are $n - 8$ cells of R_0 remaining, and we need to reserve $n - 8$ pebbles to cover them by line. That is, we have two extra pebbles. If j pebbles ($0 \leq j \leq 2$) are in the zones A_1, A_2, and A_3, then we have the following placement conditions: (a) the remaining $n - 8 + (2 - j)$ pebbles must be placed outside zones A_1, A_2, and A_3; (b) exactly $2 - j$ columns receive two pebbles, and the rest receive one pebble each. We further distinguish three subcases depending on the value of j.

Subcase 1: $j = 0$. That is, the remaining $n - 8 + 2$ pebbles are placed outside A_1, A_2, and A_3. Consider the six designated cells of R_3 in zones A_1, A_2, and A_3 [Fig. 3(d)]. Since R_3' can receive one pebble, we cannot cover these six cells entirely.

Subcase 2: $j = 1$. Call the pebble x. Note that x cannot be in the middle zone A_2. Otherwise, the four cells of R_3 in zones A_1 and A_3 cannot be covered. Suppose that x is in zone A_1. The case when x is in zone A_3 is similar. To cover the four cells of R_3 in zones A_2 and A_3, we need to place two pebbles in the row below R_3 and one pebble in the row above R_3, and they must be placed in columns adjacent to zones A_2 and A_3. [See Fig. 3(d).] As a consequence, zones A_2 and A_3 must be one column apart. (Otherwise, we have a contradiction immediately.) Due to the placement of x_5 and x_6 in the same row, we can find an additional empty row R. Together with R_1 and R_2 we have three empty rows. Assume that R is the middle of the three empty rows. Thus, R must be at least two rows apart from R_3 and R_0 so that its cells cannot be covered by distance by pebbles x_2, x_3, x_5, and x_6. To cover the four designated cells a, b, c, and d of R, we will be placing pebbles in columns adjacent to zones A_2 and A_3 and violating condition (b), which allows only one column to receive two pebbles.

Subcase 3: $j = 2$. Call these two pebbles x and y. In this case we cannot place two pebbles in the same column outside the zones A_1, A_2, and A_3. Note that both x and y cannot be placed in rows adjacent to R_3; otherwise, we will need at least two more pebbles to cover the remaining designated cells of R_3 and will have to place two pebbles in the row above R_3, which is not allowed. On the other hand, if both x and y are not placed in rows adjacent to R_3, we can always find at least three cells among the six designated cells of R_3 which are not covered by x and y. These three uncovered cells require at least three pebbles x′, y′, and z′, and two of them must be placed in the row below R_3 and in columns adjacent to the corresponding zones. Because of these two pebbles being in the same row, we can find another empty row R. That is, we have three empty rows R, R_1, and R_2. Suppose R is the middle of these three empty rows. Since there are three zones A_1, A_2, and A_3, we can find at least one zone, say A_1, such that it does not contain x or y. To cover the two designated cells of R in this zone, A_1, we have to place two pebbles in columns adjacent to zone A_1, one of which is occupied by x′, y′, or z′, and violates condition (b) that no two pebbles occupy the same column.

Case 2. Eight cells are covered by distance. In this case we have two zones, as shown in Fig. 3(e). We remark that in zone A_2 of Fig. 3(e) the two pebbles covering five cells of R_0 need not be in the same row. We must reserve $n + 1 - 8 = n - 7$ pebbles to cover by line the remaining cells of R_0. In other words, we have one extra pebble. But since cell d as shown cannot be covered by pebbles outside the zones, the extra pebble must be placed in zone A_2 to cover it. Thus we have the same placement conditions that the remaining pebbles are to be placed outside the zones and no two pebbles outside the zones occupy the same column. Let us call the extra pebble x. If x is used to cover the three designated cells in zone A_2, then to cover the remaining designated cells we need to place two pebbles in the row below R_3, thereby creating an additional empty row. Thus, we have three empty rows, and cell d′ of the middle empty row R in zone A_2 cannot be covered, a contradiction. But if x is used to cover by line, it must be used to cover cell d. To cover the designated cells of R_3, we have to place two pebbles in the row below R_3. Following the same line of reasoning, we will find two pebbles placed in a column

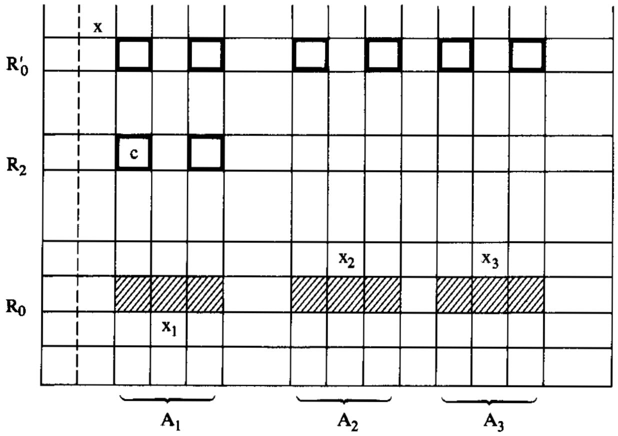


Figure 4 Illustration for the proof of Lemma 7.

adjacent to the zones, which violate the placement condition.

Case 3. Seven cells are covered by distance. Since we have $n - 6$ cells to be covered by line, we do not have extra pebbles. We therefore have the placement condition that the remaining pebbles are placed outside zones A_1 and A_2, and no two pebbles are in one column. Considering the four designated cells of row R_3 [Fig. 3(f)] and following the arguments given earlier, we will be forced to violate the placement condition.

• *Corollary 4*

$v(n, n + 2) > n - 3, \quad n \geq 7.$

Proof

By Lemma 2, $v(n, n + 2) \geq v(n, n + 1) > n - 3$.

• *Corollary 5*

$v(5, 7) > 3.$

Proof

By contradiction. Suppose three pebbles are sufficient. There are at least two empty rows, R_1 and R_2. Consider row R_1. It must be covered by distance by two pebbles, x_1 and x_2. Pebbles x_1 and x_2 cover six cells of R_1, and the remaining cell is covered by line by the last pebble. Thus, we can find an empty cell in row R_2 which is left uncovered—a contradiction.

• *Corollary 6*

$v(6, 8) > 4.$

Proof

Similar to the proof of Corollary 5.

• *Lemma 7*

$v(n, n + 3) > n - 2, \quad n \geq 3. \quad (10)$

Proof

By contradiction. Suppose $n - 2$ pebbles are sufficient. There are at least two empty rows and five empty columns. By arguments similar to those used in proving (1) and (2) of Lemma 3, we conclude that

1. The bottom row cannot be empty, and
2. If the bottom i rows are nonempty and the $(i + 1)$th row is empty, then none of these i nonempty rows receives more than one pebble.
3. Now, we consider the first empty row from the bottom, R_0. If R_0 is covered by distance by only two pebbles, then there are at least $(n + 3) - 6 = n - 3$ cells of R_0 to be covered by line. It is rather obvious that $n - 2$ pebbles will never be sufficient. In fact, all the empty rows must be covered by distance by at least three pebbles. For $n = 3$ or 4, we have at most two pebbles available, and a contradiction results.

Thus, assume that we have a configuration as shown in Fig. 4 and that these three pebbles, x_1, x_2, and x_3, cover by distance nine cells of R_0. We therefore need to reserve $n + 3 - 9 = n - 6$ pebbles to cover by line the remaining cells of R_0, and there is only one extra pebble. Since we have at least two more empty rows, let us consider the topmost empty row R'_0, and its six designated cells. It is easy to show that when the remaining $n - 5$ pebbles are placed outside the zones A_1, A_2, A_3 we cannot cover these six cells of R'_0 by using three pebbles. We therefore assume that the extra pebble is inside the zones. The extra pebble can cover at most two cells of R'_0, so we need to place two pebbles in the row below R'_0 and one (which is the only possibility) pebble above R'_0 to cover the remaining four cells of R'_0. As a result we have at least two more empty rows, R_1 and R_2. Notice that the pebble placed in the row above R'_0 must be in a column adjacent to the zones A_1, A_2, or A_3. Without loss of generality assume that it is placed in a column to the left of zone A_1, as shown in Fig. 4. Now consider the empty row R_2, which is the first empty row below R'_0, and the cell c as shown. To cover cell c we must place a pebble in the column occupied by x, thus violating the placement condition that no two pebbles outside the zones are placed in the same column. In the case when the three pebbles x_1, x_2, and x_3 cover fewer than nine cells of R_0, we can easily argue that a contradiction will result.

• *Corollary 8*

$v(n, n + 4) > n - 2, \quad n \geq 5. \quad (11)$

• *Remark*

By Lemma 7, we can in fact prove that $v(n, n + 4) > n - 2$ for $n \geq 3$. But for $n = 3$, 4, we have a better bound as follows.

• *Corollary 9*

$v(3, 7) > 2, \qquad v(4, 8) > 3.$

Proof

Similar to Lemma 7.

• *Lemma 10*

$$v(n, n + 5) > n - 1 \text{ for } n \geq 1. \tag{12}$$

Proof

By contradiction. Suppose $n - 1$ pebbles are sufficient. There exists at least one empty row, R_0, and six empty columns. By similar arguments we have

1. The bottom row must not be empty;
2. If the bottom i rows are nonempty and the $(i + 1)$th row is empty, then none of these i rows receives more than one pebble.

The proof of the lemma follows.

Suppose R_0 is the first empty row from the bottom and is covered by three pebbles x_1, x_2, and x_3, as shown in Fig. 5. These three pebbles must cover exactly nine cells of R_0; otherwise we will have a contradiction immediately. Since we have $n - 4$ pebbles available and exactly $n - 4$ cells of R_0 to be covered by line, the remaining pebbles must be placed outside A_1, A_2, and A_3, and each column receives exactly one pebble. Due to the placement of two pebbles in the row above R_0, we have at least one more empty row, R'_0. Assume that it is the topmost empty row. As before, consider the six designated cells of R'_0 in zones A_1, A_2, and A_3. First note that R'_0 cannot be one row apart from R_0. Otherwise the two designated cells in zone A_1 require that two pebbles be placed in the row above R'_0, which is not allowed for the same reason that we cannot have two pebbles in the row below R_0. Note that R'_0 is not one row apart from R_0 and the rows adjacent to R'_0 can receive at most three pebbles; it is easy to see that the three pebbles cannot cover the six designated cells of R'_0.

• *Corollary 11*

$$v(n, m) > n - 1 \text{ for } m > n + 5. \tag{13}$$

Proof

By Lemma 2.

• *Lemma 12*

$$v(n, n) > n - 4, \qquad n \geq 7. \tag{14}$$

Proof

By induction. Assume that it is true for n. Then we have

$v(n + 1, n + 1) > v(n, n + 1)$ by Lemma 2.

$v(n, n + 1) > n - 3 = (n + 1) - 4$ by Lemma 3.

Thus, $v(n, n) > n - 4$.

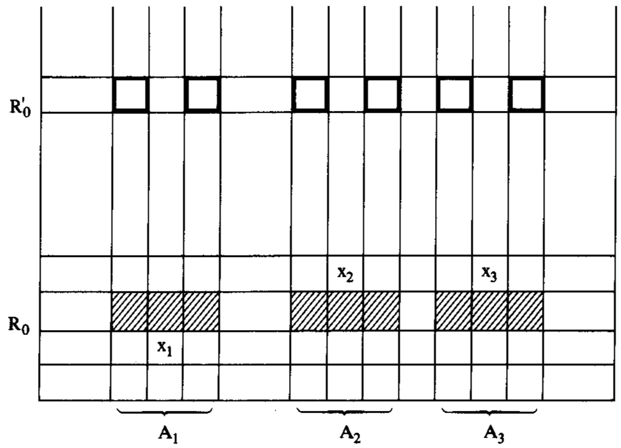


Figure 5 Illustration for the proof of Lemma 10.

4. Realization of lower bounds for *v(n, m)* under condition (a)

Next, we show how these lower bounds can be achieved by exhibiting explicit construction rules.

For example, by Lemma 3, we have $v(n, n + 1) > n - 3$, *i.e.*, $v(n, n + 1) \geq n - 2$ for $n \geq 5$. It suffices to give a placement for an $n \times (n + 1)$ board requiring exactly $(n - 2)$ pebbles. To do this, we start with a specific construction for $n = 5$ [Fig. 6(a)]. Inductively, if we already have a placement for an $n \times (n + 1)$ board requiring $(n - 2)$ pebbles, then we can put this in one of the four corners of an $(n + 1) \times (n + 2)$ board and one more pebble in one of the three remaining corner cells so that this new pebble is not in conflict with the $(n - 2)$ pebbles already placed. An example is given in Fig. 6(b). Consequently, we have a placement for an $(n + 1) \times (n + 2)$ board with $(n - 1)$ pebbles.

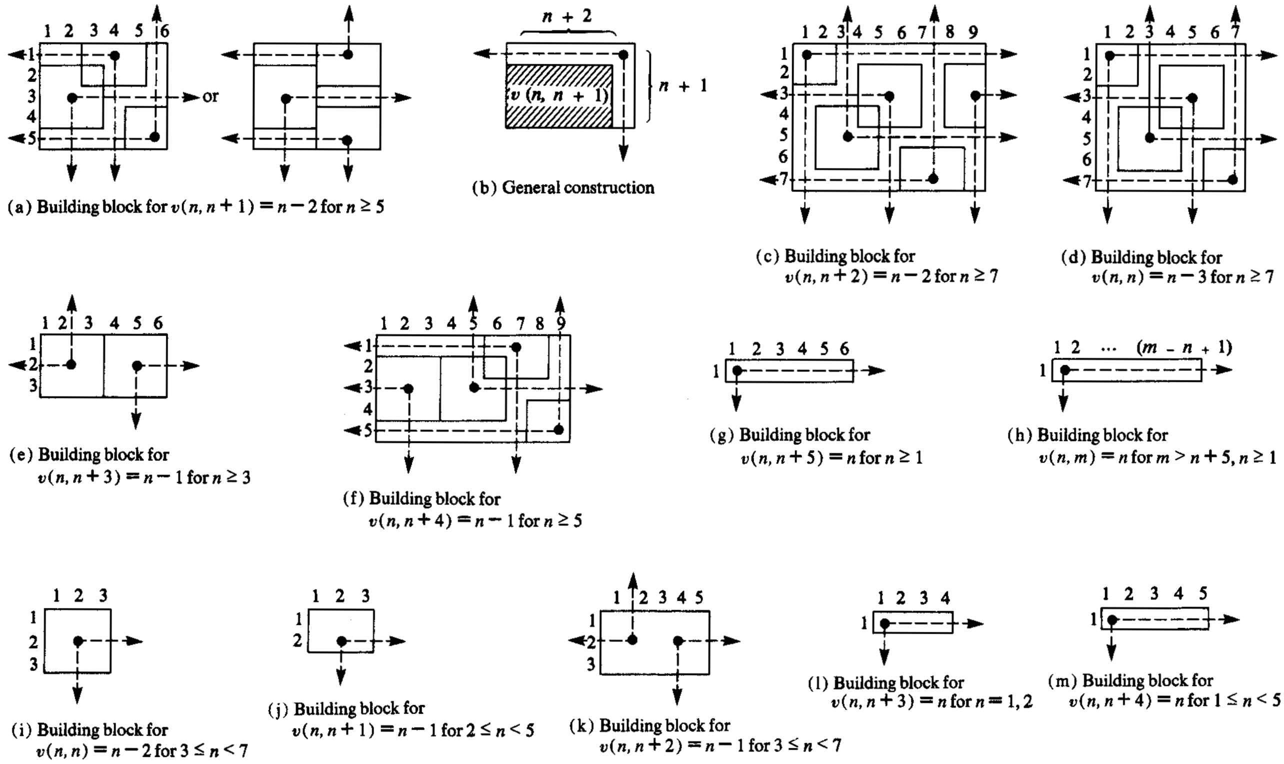
Similar construction works for all the other lemmas [Figs. 6(c)–6(h)].

Finally, we display the building blocks for $v(n, n), \cdots, v(n, n + 4)$ for small values of n in Figs. 6(i) through 6(m). For all these cases, it is easy to check that the number of pebbles placed is indeed minimum.

All these results are shown in Table 1.

5. The number *V(n, m)*

We prove that $V(n, m) = 2 \min (n, m)$ if $\max (n, m) \geq 9$. Recall that $V(n, m) \leq 2 \min (n, m)$ is always true. Thus, as long as we can give a construction where the number of pebbles is equal to $2 \min (n, m)$, it is maximum.



(a) Building block for $v(n, n+1) = n-2$ for $n \geq 5$

(b) General construction

(c) Building block for $v(n, n+2) = n-2$ for $n \geq 7$

(d) Building block for $v(n, n) = n-3$ for $n \geq 7$

(e) Building block for $v(n, n+3) = n-1$ for $n \geq 3$

(f) Building block for $v(n, n+4) = n-1$ for $n \geq 5$

(g) Building block for $v(n, n+5) = n$ for $n \geq 1$

(h) Building block for $v(n, m) = n$ for $m > n+5, n \geq 1$

(i) Building block for $v(n, n) = n-2$ for $3 \leq n < 7$

(j) Building block for $v(n, n+1) = n-1$ for $2 \leq n < 5$

(k) Building block for $v(n, n+2) = n-1$ for $3 \leq n < 7$

(l) Building block for $v(n, n+3) = n$ for $n = 1, 2$

(m) Building block for $v(n, n+4) = n$ for $1 \leq n < 5$

Figure 6 Construction rules and building blocks for specific realizations.

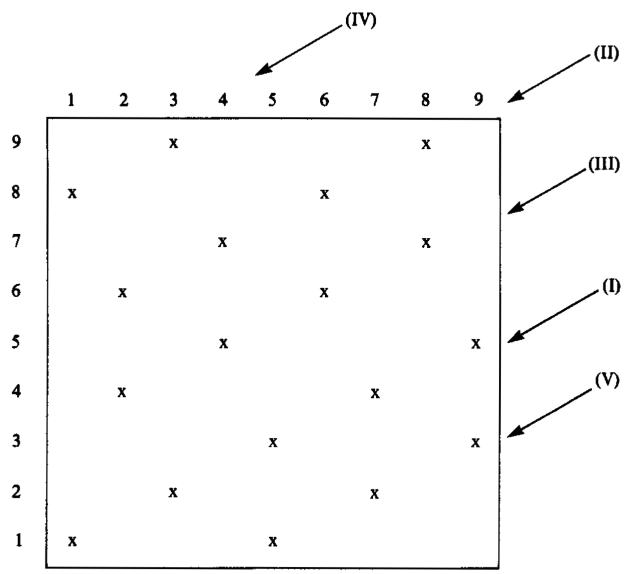


Figure 7 Construction with $n = 9$.

• *Lemma 13*

$V(9, 9) = 18$.

Proof

See the construction in Fig. 7.

• *Corollary 14*

$V(n, 9) = 2n$ for $n \leq 9$.

Proof

Take a submatrix of dimension $n \times 9$ of the 9×9 matrix displayed in Fig. 7. Since each row has exactly two pebbles, the total is $2n$.

• *Lemma 15*

$$V(n, m) = 2 \min(n, m), \text{ if } \max(n, m) \geq 9. \quad (8)$$

Proof

Without loss of generality, we can assume $m \geq n$. Thus $m \geq 9$.

Case 1 If $n < 9$, then we can consider the first 9 columns and apply Corollary 14.

Case 2 $n \geq 9$. It suffices to consider the square matrix $n \times n$.

1. n is odd. The construction is similar to the one in Fig. 7, where $n = 9$. Specifically, let us number the rows from the bottom up and the columns from left to right. We use i to index the rows and j to index the columns. Then we place the pebbles one at a time. In general, if the last pebble is in location (i, j), then we place the next pebble in location (i', j'), where

$$i' = (i+1) \mod n,$$
$$j' = (j+2) \mod n.$$

Table 1 Values of $v(n, m)$ under neighboring condition (a): $v(n, m) = v(m, n)$.

n \ m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2		1	1	2	2	2	2	2	2	2	2	2	2	2	2	2
3			1	2	2	2	3	3	3	3	3	3	3	3	3	3
4				2	3	3	3	4	4	4	4	4	4	4	4	4
5					3	3	4	4	4	5	5	5	5	5	5	5
6						4	4	5	5	5	6	6	6	6	6	6
7							4	5	5	6	6	7	7	7	7	7
8								5	6	6	7	7	8	8	8	8
9									6	7	7	8	8	9	9	9
10										7	8	8	9	9	10	10
11											8	9	9	10	10	11
12												9	10	10	11	11
13													10	11	11	12

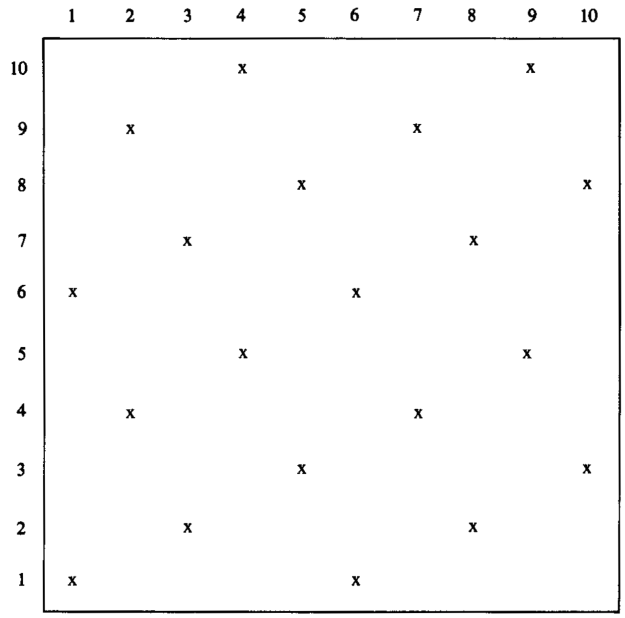


Figure 8 Construction with $n = 10$.

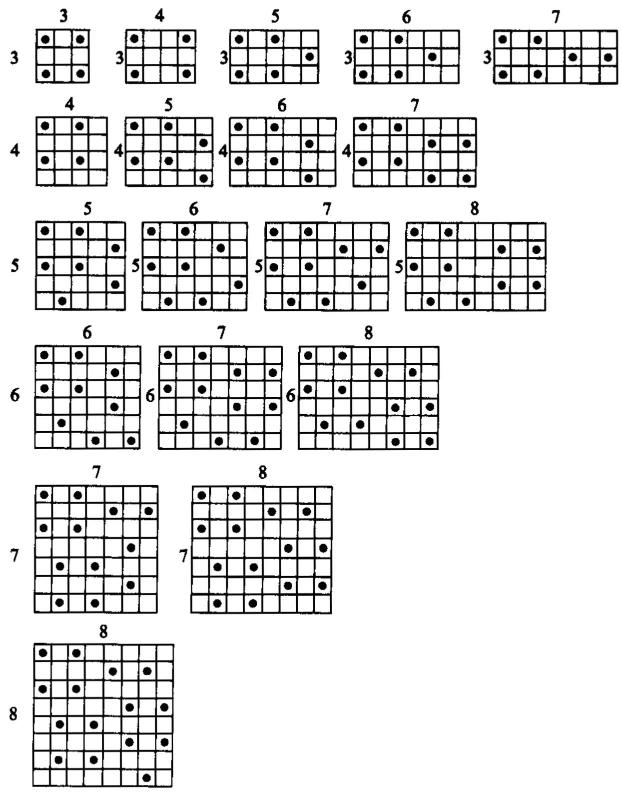


Figure 9 Placement achieving $V(n, m)$ for $n, m < 9$.

The construction has two phases:

Phase 1—Place a pebble at location (1, 1) as a starting point, and continue the process until n pebbles have been placed.

Phase 2—Place a pebble at location (4, 2) as a new starting point, and continue the process until n pebbles have been placed.

Referring to Fig. 7, we have generated in Phase 1 two "strings" of pebbles: (I) and (II). String (I) starts at (1, 1); string (II) starts at $(\lceil n/2 \rceil + 1, 2)$. Phase 2 has generated three strings: (III), (IV), and (V). String (III) starts at (4, 2). String (IV) starts at $(\lceil n/2 \rceil + 3, 1)$. String (V) starts at $(1, n - 4)$. It is easy to check that there are exactly two pebbles in each row and two pebbles in each column, with a total of $2n$ pebbles.

It suffices to check that if a pebble is at (i, j), then its neighboring positions (at most 8) cannot have any pebbles. First, for any two pebbles at positions (i, j) and (i', j'), define their *distance* as $\max(|i - i'|, |j - j'|)$. Then it suffices to check that for any two pebbles, their distance must be at least 2. For pebbles belonging to the same string, this condition is true by construction. For pebbles belonging to different strings, we only have to look at the starting and second positions of the five strings and to check that the distance of any two of these positions is at least 2. For example, the five starting positions are (1, 1), $(\lceil n/2 \rceil + 1, 2)$, (4, 2), $(\lceil n/2 \rceil + 3, 1)$, and $(1, n - 4)$. Clearly, for $n \geq 9$, the distance between any two of them is at least 2. Similarly, we can check the ten starting and second positions for their distances.

2. n is even. The construction is exactly the same as before, except that the starting positions of the five strings are now (1, 1), $((n/2) + 1, 1)$, (4, 2), $((n/2) + 4, 2)$ and $(1, n - 4)$. (See Fig. 8 for $n = 10$.)

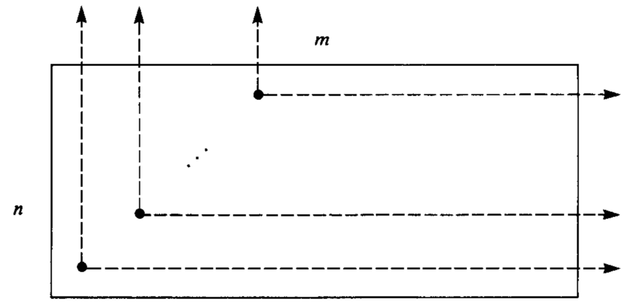


Figure 10 Arrangement of $v(n, m)$ pebbles for conditions (b), (c), (d), and (e).

Table 2 Values of $V(n, m)$ under neighboring condition (a): $V(n, m) = V(m, n)$.

n \ m	1	2	3	4	5	6	7	8	9
1	1	1	2	2	2	2	2	2	2
2		1	2	2	3	3	4	4	4
3			4	4	5	5	6	6	6
4				4	6	6	8	8	8
5					7	8	9	10	10
6						9	11	12	12
7							12	14	14
8								15	16
9									18

Table 3 Values of $v(n, m)$ under neighboring conditions (b), (c), (d), and (e).

n \ m	1	2	3	4	5
1	1	1	1	1	1
2	1	2	2	2	2
3	1	2	3	3	3
4	1	2	3	4	4

For $n, m < 9$, we have explicit constructions displayed in Fig. 9. In each case one can prove that indeed the figure contains the maximum number of pebbles. The results are shown in Table 2.

6. Other neighboring conditions

So far, we assume that when a via hole exists at position (i, j), then all its neighboring positions (at most 8) are restricted from having any via holes. In this section, we consider four other kinds of neighboring constraints (b, c, d, e of Fig. 1), namely,

(b) only positions $(i + 1, j)$, $(i - 1, j)$, $(i, j + 1)$, $(i, j - 1)$ are restricted,
(c) only $(i + 1, j)$, $(i - 1, j)$ are restricted,
(d) only $(i, j + 1)$, $(i, j - 1)$ are restricted, and
(e) no restrictions.

• *Lemma 16*
For each of the four neighboring conditions (b), (c), (d), and (e), we have $v(n, m) = \min(n, m)$.

Proof
We first prove that $v(n, m) \geq \min(n, m)$. Assume otherwise; then $v(n, m) < \min(n, m)$. Further assume $n \leq m$. Thus, $v(n, m) < n \leq m$. We shall derive a contradiction for each neighboring condition.

1. *Condition (b)*. Since $v(n, m) < n$, there must exist an empty row. By the same token, $v(n, m) < m$ implies the existence of an empty column. Let the intersection position of the empty row and empty column be a. Clearly, a cannot be covered by line or by distance with the existing pebbles, a contradiction.
2. A similar proof applies to conditions (c), (d), and (e).

We next give an explicit construction to prove that the lower bound is achievable. Indeed, just place the n pebbles along the diagonal as shown in Fig. 10, and the result follows.

The values of $v(n, m)$ are presented in Table 3.

• *Lemma 17*
For each of the neighboring conditions (b), (c), and (d), $V(n, m) = 2 \min(n, m)$ for $\max(n, m) \geq 4$. For (e), $V(n, m) = 2 \min(n, m)$ for $\max(n, m) \geq 2$.

Proof

1. *Conditions (b), (c), (d)*. Without loss of generality, assume $n \leq m$. In general, place the pebbles one by one starting with position (1, 1), *i.e.*, the lower left corner. If the last pebble is at position (i, j), then the next one is at (i', j'), where $i' = (i + 1)$ and $j' = (j + 1)$, until n pebbles have been placed. Then start with position (1, 3), and repeat the process until another n pebbles have been placed. (See Fig. 11.) It is easy to check that these $2n$ pebbles indeed satisfy all the conditions.
2. *Condition (e)*. Just place two pebbles in each row.

For n, m such that $\max(n, m) < 4$, under condition (b), the above construction also works, but now $V(n, m) < 2$

Table 4 Values of $V(n, m)$ under neighboring condition (b): $V(n, m) = V(m, n)$.

n \ m	1	2	3	4	5	6
1	1	1	2	2	2	2
2		2	3	4	4	4
3			5	6	6	6
4				8	8	8
5					10	10
6						12

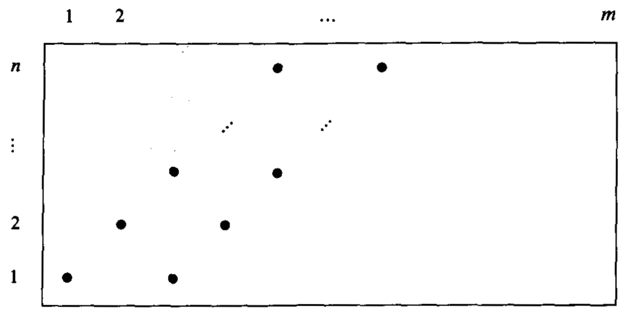


Figure 11 One maximum pebble arrangement.

Table 5 Values of $V(n, m)$ under neighboring condition (c).

n \ m	1	2	3	4	5	6
1	1	2	2	2	2	2
2	1	2	3	4	4	4
3	2	4	5	6	6	6
4	2	4	6	8	8	8
5	2	4	6	8	10	10
6	2	4	6	8	10	12

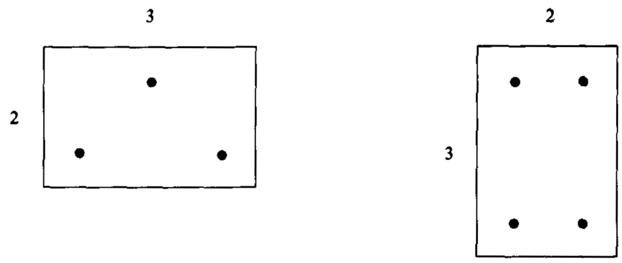


Figure 12 Maximum number of pebbles for condition (c).

min (n, m) except for $n = 1, m = 3$. In that case, $V(1, 3) = 2$. The results are given in Table 4. The same can be done for condition (c). Note that in this case $V(2, 3) = 3$ but $V(3, 2) = 4$ (Fig. 12).

The results are shown in Table 5. Also note that condition (d) just means transposing the table.

7. Acknowledgment

The authors are grateful to Dr. R. Nair and Dr. J. Lee for many useful discussions in formulating and solving the problem.

8. References

1. G. Persky, D. N. Deutsch, and D. G. Schweikert, "CTX—A System for the Directed Automatic Design of LSI Circuits," *Proceedings of the 13th Design Automation Conference*, June 1976, pp. 399–407.
2. K. H. Khokhani and A. M. Patel, "The Chip Layout Problem: A Placement Procedure for LSI," *Proceedings of the 14th Design Automation Conference*, 1977, pp. 291–297.
3. K. A. Chen, M. Feuer, K. H. Khokhani, N. Nan, and S. Schmidt, "The Chip Layout Problem: An Automatic Wiring Procedure," *Proceedings of the 14th Design Automation Conference*, 1977, pp. 298–302.
4. H. Shiraishi and F. Hirose, "Efficient Placement and Routing Techniques for Master Slice LSI," *Proceedings of the 17th Design Automation Conference*, June 2, 1980, pp. 458–464.
5. W. R. Heller, W. F. Mikhail, and W. E. Donath, "Prediction of Wiring Space Requirements for LSI," *Journal of Design Automation and Fault Tolerant Computing* **2,** 117–144 (1978).

Received January 8, 1981

Dr. Lee is located at Northwestern University, Evanston, Illinois 60201; Drs. Hong and Wong are located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.