

論 文

VLSI System CAD에 관한 研究

正會員 朴 鎮 秀*

A Study on Computer Aided VLSI System Design

Jin Soo PARK*, *Regular Member*

要 約 LSI CAD시스템에 있어서 가장 중요한 휴리스틱 레이아웃 알고리즘을 제안하고 있다. 배치 알고리즘으로서는 인간이 작성한 논리설계회로 도면상의 모듈의 위치를 그대로 배치에 반영함으로써 인간의 종합판단력을 이용한 배치방법을 제안하였다. 제안된 방법의 유용성을 보이기 위해 종래 사용되고 있는 클러스터 성장배치법과 비교하는 프로그램 실험을 행하였다. 배선 알고리즘으로서는 종래 Maze 법이 갖는 단점 즉 기억용량 과다문제를 줄이기 위한 방법을 제안했다.

ABSTRACT In this paper I have proposed a heuristic layout algorithm which is important in the CAD system of VLSI. I have designed a placement algorithm to be used the method which depends upon the synthetic judgment of human. The placement algorithm can reflect the position of a module in a logical design circuit diagram drawn up by human. Also, in order to show the usefulness of the new method I have compared through a program experiment it with the former method of cluster development placement. Moreover, a routing algorithm is proposed in order to reduce the excessive problem of memory capacity. Of course this new algorithm compensates for the former Maze's defects.

1. 서 론

VLSI의 고집적도를 가능하게 하는 기술로서는 프로세스 기술, 설계기술 등을 들 수 있으며 프로세스 기술은 머지 않은 장래에 그 한계성을 나타낼 것으로 예상된다⁽¹⁾. 따라서 설계기술이 장래의 LSI 발전에 관건이 됨은 명확관하하며 대규모 회로의 채용은 컴퓨터의 도입을 필수 불가결하게 만들 것이다. 이 기술이 현재 널리 알려져 있는 CAD 기술이며 LSI 설계에 있어 다수 채용되고 있다⁽²⁾. 특히 layout 설계는 LSI 설계에 있어서 전체 cycle 중 시간과 비용이 가장 많이 필요한 것으로서 이것에 대한 연구는 중요하다.

본 논문에서는 이러한 배경하에서 VLSI의 휴리스틱 레이아웃 (heuristic layout) 설계법을 제안한다.

즉 배치산법으로서 숙련된 인간에 의해 작성한 논리설계도로부터 도면상의 모듈의 상호 위치를 가지고 배치하는 방법으로서 설계도면 상의 모듈의 위치 및 신호선의 결합도를 산출하여 주어진 칩상에 할당하는 방법과 새로운 평가함수를 제안한다. 또한 제안된 배치산법의 우수성을 보이기 위해 종래 사용되어 왔던 클러스터 (cluster) 성장법⁽³⁾과 비교하는 프로그램 실험을 행하였다.

배선산법으로서 종래 Maze 법⁽⁴⁾을 개량한 방법을 제안한다. 즉 Maze 법은 신호선의 경로탐색이 용이하고 경로가 존재하면 반드시 찾아내는 탐색능력이 있으나 과다한 컴퓨터 기억용량이 필요하다라는 단점을 가지고 있다. 따라서 본 논문에서는 칩의 트랙들을 9개의 트랙으로 나누어 경

* 清州大學校理工大學電子工學科
Dept. of Electronic Engineering, Cheongju University
Cheongju, 310 Korea
論文番號 : 83-05 (接受 1983. 1. 22)

모를 탐색하므로써 기억용량을 대폭 감축시킬 수 있는 방법을 제안한다. 즉 나누어진 9트랙의 내부상태를 3트랙 직진경로로 코우드화하여 6개의 bit로 표현함으로써 기억용량의 절감을 각축하였다.

2. 준 비

본 논문에서 취급하는 레이아웃 문제는 크기가 정해진 2차원 칩상에 모듈을 배치 배선하는 것이다. 구체적인 대상으로서는 게이트 array LSI이며 프린트 기관의 IC배치에도 적용가능하다.

레이아웃 문제를 컴퓨터를 이용하여 해결하는 경우 물리적인 성질을 반영하면 반영할수록 계산시간은 증가한다. 따라서 본 논문에서는 문제의 본질을 변화시키지 않는 범위 내에서 다음과 같이 간략화하여 해를 구한다.

(1) 모듈(배치하고자 하는 최소단위)의 핀의 위치는 무시하며 신호선(같은 전위점을 갖는 핀간의 접속선)은 모듈의 중심에 연결하는 것으로 한다.

(2) 셀(모듈의 배치위치)은 칩(셀들의 집합)상에 규칙적으로 가로와 세로로 나열되어 있고 접속된 셀 간의 거리는 1로 하며 신호선은 트랙에 배선 접속된다.

(3) 1개의 셀에는 단 1개의 모듈이 할당되는 것으로 한다.

(4) 모듈의 크기는 전부 같은 크기로 한다.

(5) 모듈의 수와 셀의 수는 같은 것으로 한다. 일반적으로 모듈의 수는 셀의 수보다 적으나 여기서는 신호선을 갖지 않는 모듈을 정의하여 사용한다.

(6) 신호선의 접속규칙으로서 배선을 행하는 모듈 상의 순서에 의하지 않으며 계산시간이 짧은 완전Graph법⁽⁵⁾을 사용한다.

3. 배치산법

3.1 순서결정

인간에 의해 작성된 논리 설계도를 주어진 칩상에 충실히 재현하는 방법을 생각한다. 논리설계도면 상의 모듈이 전부 4상한에 존재하도록 직교 좌표계 $x-y$ 를 결정하고 모듈의 중심의 위치로부터 y 축에 평행한 직선을 긋고 x 축과의 교점에 원점에서 가까운쪽으로부터 (+)방향으로 1, 2, ..., N 을 순서적으로 붙인다. 마찬가지로 모듈의 중심의 위치로부터 y 축에 평행한 직선을 긋고

y 축과의 교점에 번호 1, 2, ..., M 을 붙인다.

3.2 압 축

주어진 칩이 $(n \times m)$ 개의 셀로 구성되어 있다고 가정한다. 통상 $N > n$, $M > m$ 으로 되지만 만약 $N < n$ 이면 $N = n$ 이 될 때까지, $M < m$ 이면 $m = M$ 이 될 때까지 가(假)배치의 칩을 확장한다. 이와 같이 하여 얻어진 가배치의 칩을 x 방향 또는 y 방향으로 압축하면서 목표로 하는 칩의 크기로 하여 최종적인 배치를 얻는다. 이하 그 방법에 대하여 논한다.

가배치 상태의 칩 크기 N, M 과 최종적으로 배치하게 되는 칩의 크기 n, m 과의 비 r_x, r_y 를 다음 식으로 구한다.

$$r_x = \frac{N}{n}, \quad r_y = \frac{M}{m} \quad (1)$$

r_x 와 r_y 의 크기를 비교하여 $r_x > r_y$ 이면 y 방향으로부터 압축을 시작한다. 이것은 목적으로 하는 칩의 크기와의 차이가 큰 쪽을 목적으로 하는 칩의 크기에 가깝게 하려고 하는 것이다.

압축은 인접한 2행(行) 또는 2열(列)을 융합하여 1행 또는 1열로 하는 것을 의미하며 1회 압축에 의해 N 또는 M 은 1만큼 감소한다. 여기서 압축하는 방향을 x 방향으로 하면 다음과 같이 구할 수 있다.

i 열과 $(i+1)$ 열을 압축할 때 압축에 관여하는

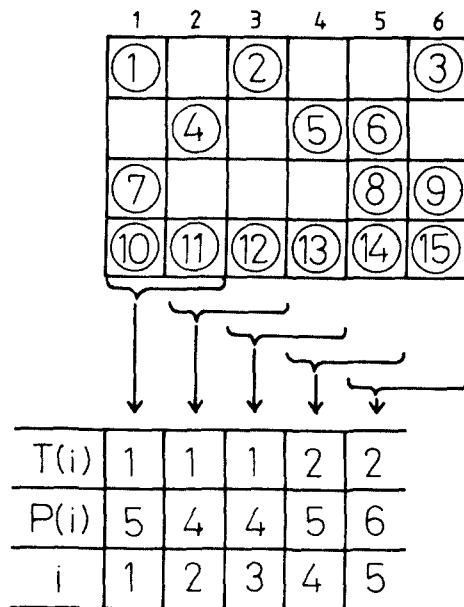


그림 1 $P(i), T(i)$ 의 Table
The table of $P(i), T(i)$.

모듈의 수 $p(i)$ 와 압축했을 때 2개의 모듈이 동일 셀을 점유하게 되는 상태(이것을 “타이”라 부른다)의 갯수 $T(i)$ 를 구한다. 그림 1에 한 예를 나타냈다.

최소치의 $T(i)$ 를 갖는 i (그림 1에서는 $i=1, 2, 3$)에 대하여 압축을 행한다. 이 때 $T(i)$ 의 최소가 되는 i 가 복수개 존재하면 최소의 $p(i)$ 를 갖는 i 에 관해 압축을 행한다. 만일 $p(i)$ 가 복수개 존재하면 임의로 선택하는 것으로 한다. 또 압축한 위치의 $T(i)$ 가 0 즉 타이가 존재하지 않으면 다시 압축방향과 압축위치를 결정해 압축을 반복한다.

3.3 모듈의 이동

타이가 존재할 때 타이가 된 모듈의 위치로부터 직각거리(맨허턴 거리)로 제일 가까운 빈 셀을 찾아 모듈을 이동한다. 복수개의 빈 셀이 타이의 위치로부터 동일한 거리에 있는 경우에는 임의적으로 선택하는 것으로 한다. 그림 2는 압축한 결과 A, B가 타이 모듈이 된 경우로 빈 셀은 1개 밖에 없다고 가정한다.

이 경우 좌표 (i, j) 에 있어 모듈 A, B의 셀로부터 가장 가까운 빈 셀의 좌표가 $(i+k, j+l)$ 이다. 타이 모듈의 이동은 최단거리로 행하여지기 때문에 가상 칩 상의 4점 (i, j) , $(i, j+l)$, $(i+k, j)$, $(i+k, j+l)$ 으로 포함될 범위 이외에는 이동에 관계없는 셀이 된다. 그림 3.3에 (i, j) 에 존재한 타이 모듈을 이동시켰을 때 네 가지 형태를 나타내었다.

이 네 가지 직각의 경우에 관하여 $A \cdot B$ 의 평가 함수 $F(A)$, $F(B)$ 를 계산하여 코스트를 계산한다.

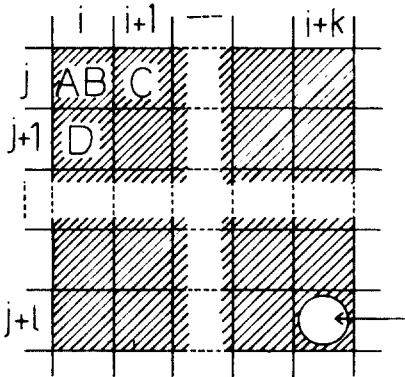


그림 2 타이 모듈의 이동범위
The transfer region of tie-module.

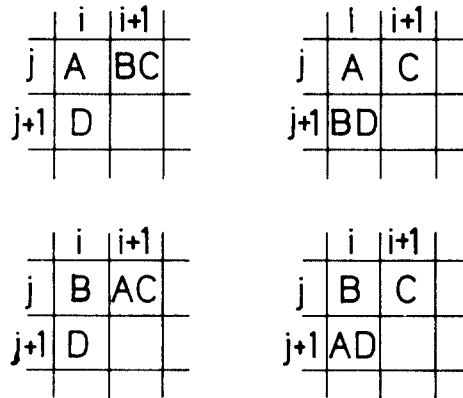


그림 3 타이 모듈의 이동방법
The transfer method of tie-module.

$$F_{tie}(A \cdot B) = F(A) + F(B) \quad (2)$$

위 식의 값이 최소가 되는 점으로 모듈을 이동하여 타이를 해결한다.

3.4 평가함수

압축과정에 있어 발생하는 타이 모듈에 대하여 이동하는 방향을 결정하기 위한 평가함수를 정의한다. 사용되는 기호는 다음과 같다.

k : 모듈의 총수

$L_{x(i)}(L_{y(i)})$: 설계도 상에서의 모듈 i 로부터 j 까지의 $x(y)$ 방향의 거리

$l_{x(i)}(l_{y(i)})$: 배치한 칩 상에서의 모듈 i 로부터 j 까지의 $x(y)$ 방향의 거리

θ_{ij} : 설계도 상에서의 모듈 i 로부터 j 를 보았을 때의 각도(x 축을 기준)

θ'_i : 배치한 칩 상에서의 모듈 i 로부터 j 를 보았을 때의 각도(x 축을 기준)

r_{ij} : 모듈 i 와 j 사이의 신호선수

$\Delta\theta_{ij} = |\theta_{ij} - \theta'_i|$: 각도의 재현도

만약 칩 상에서의 배치결과와 설계도 상의 $\Delta\theta_{ij}$ 값이 적으면 원래 설계도를 충실히 재현한 것으로 된다. 그러나 배치를 압축하여가면 $\Delta\theta_{ij}$ 는 변화하여 가므로 재현도가 큰 배치를 얻기 위하여는 $\Delta\theta_{ij}$ 에 그림 4와 같은 모양의 비선형단조증가함수를 적용시켜 다음과 같이 $\Delta\theta'_{ij}$ 를 도입한다.

$$\Delta\theta'_{ij} = f(\Delta\theta_{ij}) \quad (3)$$

모듈 i 와 j 사이의 거리는 맨허턴 거리로 표현되나 설계도 상에서의 거리와 배치완료한 칩 상에서의 거리를 비교할 때 문제가 된다. 칩 상에서의 최종배치가 얻어질 때까지는 항상 압축되는 크기의 비율이 변화되기 때문이다. 그러므로

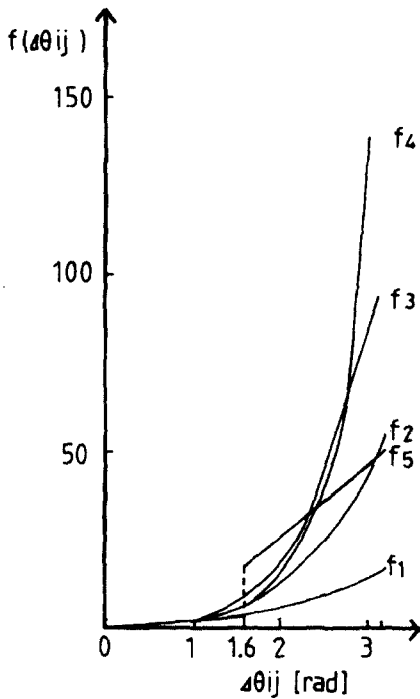


그림 4 각도의 제편도 함수
The ΔA_{ij} function of angle.

여기서는 양자의 거리 축소비를 같이 하는 의미로 설계도와 압축결과에 있어 1변의 길이가 1인 정사각형으로 기준화하여 취급하기로 한다. 기준화한 길이를 *표를 붙여 다음과 같이 정의한다.

$$l_{xu}^* (l_{yu}^*) = l_{xu} / N (l_{yu} / M) \quad (4)$$

$$l_{xu}^* (l_{yu}^*) = l_{xu} / n (l_{yu} / m) \quad (5)$$

따라서 모듈 i 와 j 사이의 거리는 각각

$$l_{ij}^* = l_{xu}^* + l_{yu}^* \quad (6)$$

$$l_{ij}^* = l_{xu}^* + l_{yu}^* \quad (7)$$

로 되며 기준화된 거리의 비를 다음 식으로 결정한다.

$$R_{ij} = l_{ij}^* / l_{ij} \quad (8)$$

또한

$$G(R_{ij}) = R_{ij} + 1/R_{ij} \quad (9)$$

의 함수를 도입한다. 이것은 $R_{ij} = 1$ 즉 설계도와 압축결과와 기준화한 거리가 같을 때 최소치를 취한다. $R_{ij} < 1$, $R_{ij} > 1$ 모두 같은 비율로 G 는 증가한다.

이상의 정의된 함수를 사용하여 모듈 i 의 평가함수를 다음과 같이 정의한다.

$$F(i) = \sum_{j=1}^k [f(\Delta\theta_{ij}) \cdot G(R_{ij}) \cdot r_{ij}] \quad (10)$$

$F(i)$ 는 설계도에 있어서 다른 모듈에 대한 상대적인 위치와 배치 칩 상에서의 상대적인 위치차와 떨어진 거리비를 고려한 함수이다.

더우기 배치결과 전체의 평가함수로서 코스트를 다음과 같이 정의한다.

$$C = \frac{1}{k} \sum_{i=1}^k F(i) \quad (11)$$

3.5 프로그램 실험 및 고찰

제안한 알고리즘을 프로그래머 실험을 행하였다. 사용한 컴퓨터는 UNIVAC 9030이며 FORTRAN으로 작성하였다. 또한 CPU Time = 2.314sec이며 flowchart는 부록과 같다.

지금까지 초기배치 알고리즘으로서 널리 사용되고 있는 클러스터 성장법⁽³⁾을 적용하여 얻은 배치결과와 본 방법에 의한 배치결과를 총 배선장 및 최대 cut수를 비교검토하였다.

2개의 논리 설계도를 사용했으며⁽⁶⁾ 예제 1은 43개의 모듈로 구성된 실제 회로도에 2개의 더미 모듈을 정의하여 $m=9$, $n=5$ 의 셀로 구성된 칩에 배치했다. 예제 2는 36개의 모듈로 구성된 실제 회로도를 $m=n=6$ 의 셀로 구성된 칩에 배치했다. 그 결과를 표 1에 나타냈다.

단, 클러스터 성장법을 적용할 때 $\lambda=0$ 로 하였다.

표 1에서 알 수 있듯이 제안한 배치방법이 총 배선장 및 최대 cut수를 배치평가 목적함수로 했을 때 클러스터 성장법보다 충분히 좋은 결과를 얻을 수 있었다.

4. 배선산법

4.1 9트랙 직진경로

9트랙 방식을 이용한 배선경로 탐색법은 가로로 3열의 트랙, 세로로 3행의 트랙의 정사각형이 되는 9개의 트랙을 기본단위로 하여 배선을 행하는 것이다(이하 9트랙의 기본 트랙을 대트랙이라 부르고 기본 트랙을 구성하고 있는

표 1 배치 결과의 비교
The contrast of placement result.

평가목적함수	예제	본방법	클러스터 성장법
총 배선장	(I)	72	116
	(II)	90	100
최대 cut 수	(I)	10	18
	(II)	13	14

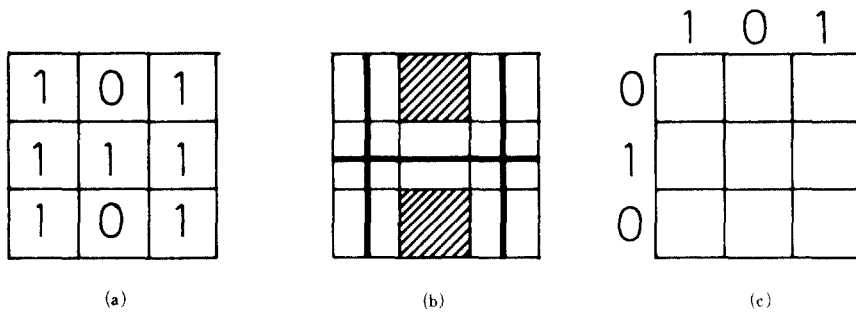


그림 5 3트랙 직진경로
3-track directed path.

요소를 소트랙이라 부른다).

그림 5에 트랙을 나타냈으며 (a)에서는 배선 불가능한 소트랙에는 0, 배선가능한 소트랙에는 1을 할당한 것을, (b)에서는 (a)로부터 3개의 소트랙을 직진하는 경로는 3개 이외에는 없음을 나타냈다. 따라서 직진가능한 경로를 찾아내기 위해서는 직진 가능한 행이나 열에는 1을, 불가능한 것에는 0를 할당함으로써 그것을 그림 5(c)에 나타내었다.

직진 가능한 경로의 패턴을 그림 6에 열거하였다. 논리적으로 행 bit와 열 bit를 각각 3개씩 사용하므로 전체적으로 $2^3 \times 2^3 = 64$ 개의 패턴이 존재하게 된다.

이 패턴 중 $(1, 1, 1)X_x$ 와 같은 형은 16개 존재하지만, $(1, 1, 1)X_x$ 를 제외하고 남은 것은 논리적으로 모순이 된다.

따라서 나머지는 50개의 패턴이 남게 되며 이중 34개의 패턴은 9트랙의 내부상태를 표현할 수 있으나 (그림 6에서 $(1, 1, 1)X_x$ 와 빗금친 것이 포함된 것) 나머지 16개의 패턴은 9트랙의 내부상태를 표현할 수 없다. 그런데 이 16개의 패턴은 공통성을 가지므로 다음과 같은 공통성을 갖는 유형을 3가지 (P_1, P_2, P_3)로 분류하기로 한다.

- (1) P_2 형 : 행과 열에 1개씩 3트랙 직진경로가 존재하는 경우 (9개 패턴)
- (2) P_1 형 : 행 혹은 열에 1개만 3트랙 직진경로가 존재하는 경우 (6개 패턴)
- (3) P_0 형 : 행과 열에 3트랙 직진경로가 전혀 없는 경우 (1개 패턴)

그 외에 행이나 열에만 2개의 3트랙 직진경로가 존재하는 경우에는 9트랙의 내부상태를 결정할 수 있으며 P_2, P_1, P_0 순으로 3트랙 직진경

로가 감소함을 알 수 있다.

4.2 알고리즘

S (start)와 T (terminal)을 포함한 대트랙 S, T를 취하여 S · T에 인접한 소트랙에 경로가 존재하면 1, 그렇지 않으면 0를 할당한다. 그 다음에 인접한 소 셀간에 논리AND를 취하여 확장해 나감으로써 경로를 찾는다. 알고리즘의 구성은 다음과 같다.

스텝 1 : (초기설정)대트랙 및 소트랙을 결정한다.

스텝 2 : 소트랙에 1, 0를 할당하며 같은 행 및 열 트랙의 3트랙 직진경로를 구한다 (9트랙 코우드 결정).

스텝 3 : 패턴을 설정한다.

스텝 4 : 대트랙 간의 인접 셀에 대하여 논리AND를 취하고 배선탐색 가능성을 결정하고 탐색가능하면 스텝 4를 반복하고 그렇지 않으면 중지한다.

5. 결 론

본 논문에서는 VLSI 설계에 있어서 휴리스틱 레이아웃 알고리즘을 제안했다.

배치 알고리즘으로서는 숙련된 인간이 작성한 논리 설계도가 회로의 물리적인 성격을 반영하고 있다고 간주하여 설계도면 상의 모듈의 위치를 실제의 배치에 반영하는 방법을 제안하였다.

프로그램 실험에 의하면 종래 가장 많이 사용되어 왔던 배치평가 목적함수인 총배선장 및 최대cut수의 면에서 충분히 가치있는 것으로 판명되었다.

배선 알고리즘으로서는 종래의 Maze법이 갖는 기억용량의 과다문제를 감소할 수 있는 새로운 방법을 제안하였다.

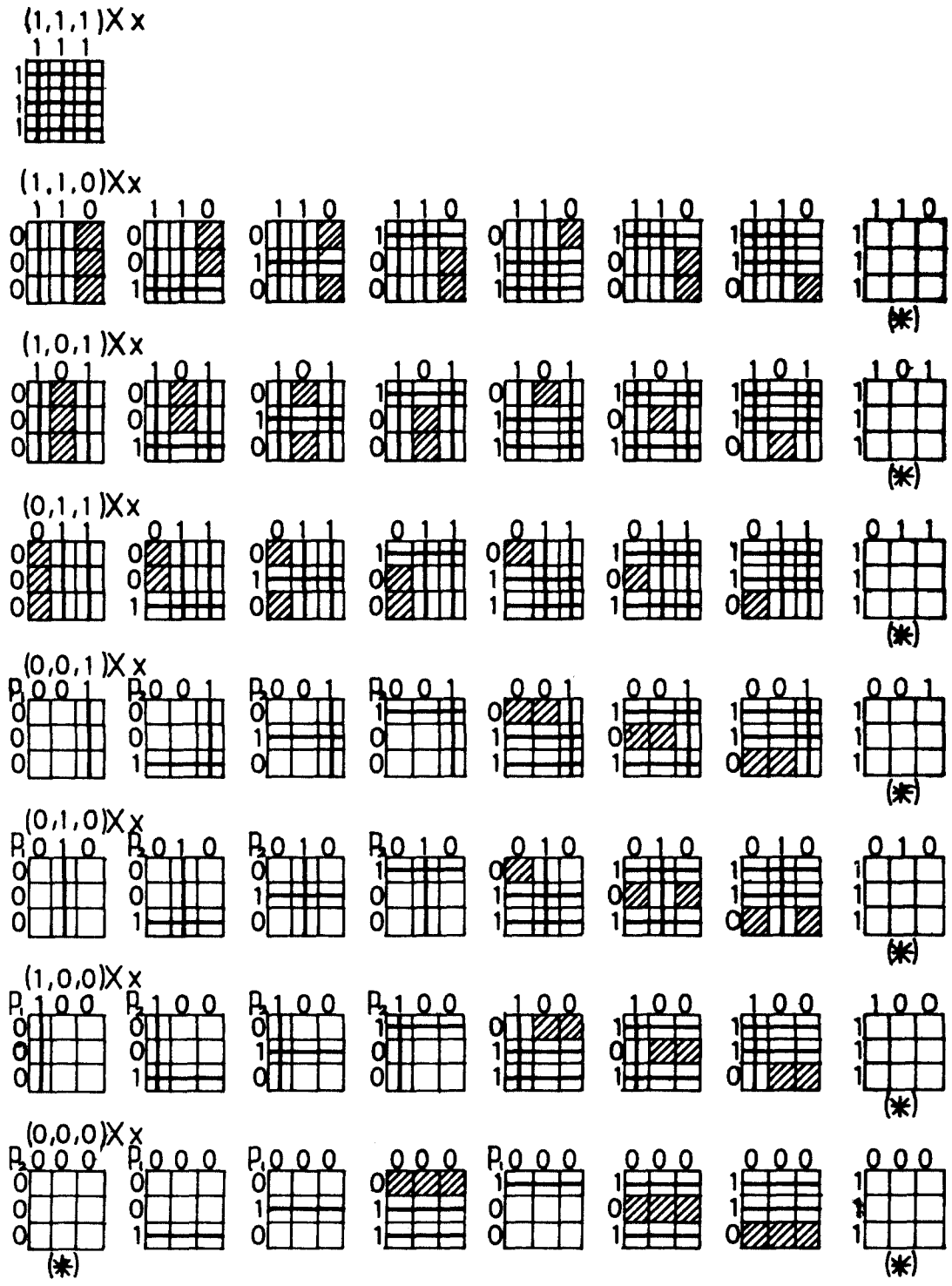


그림 6 3트랙 직진경로 패턴
3-track directed path pattern.

일반적으로 자동 배선기에서는 여러 종류의 배선 알고리즘이 혼합되어 사용되고 있으나 탐색 능력면에서는 Maze법이 탁월하므로 배선의 최종 단계에서 사용되어 왔다. 따라서 본 방법의 제안은 배선시간 및 기억용량을 감소할 수 있기 때문에 전체의 레이아웃 코스트를 감소시킬 뿐만 아니라 LSI생산 코스트를 절감할 수 있으리라 기대된다.

그러나 배치 알고리즘에 있어서 인간의 판단력에는 애매모호한 문제가 개재되기 때문에 논리설계도에 따라서는 적용하기 힘든 경우도 있으리라는 문제점이 있고 배선 알고리즘에 있어서는 휴리스틱한 방법이므로 특수한 경우 경로가 있음에도 불구하고 탐색하지 못하는 단점을 가지고 있다.

謝 辭

本研究는 文敎部로부터 1982年度 交流敎授 계획에 의한 學術研究助成費를 支援받아 行한 것 입니다.

부 록

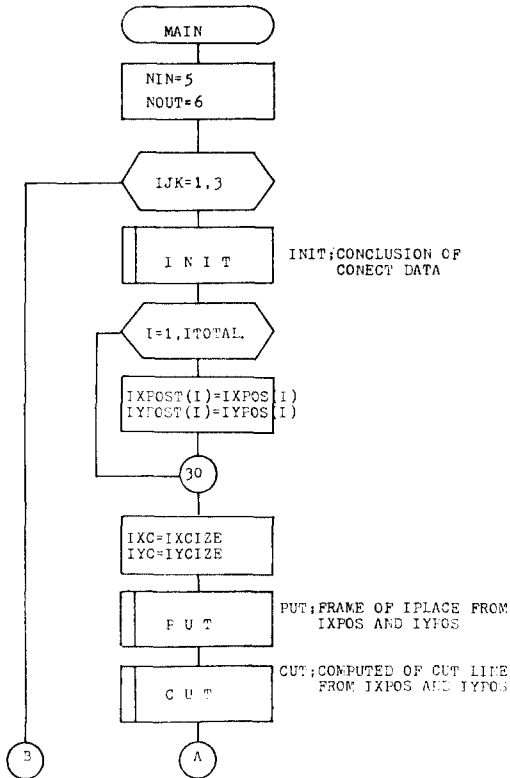


그림 1

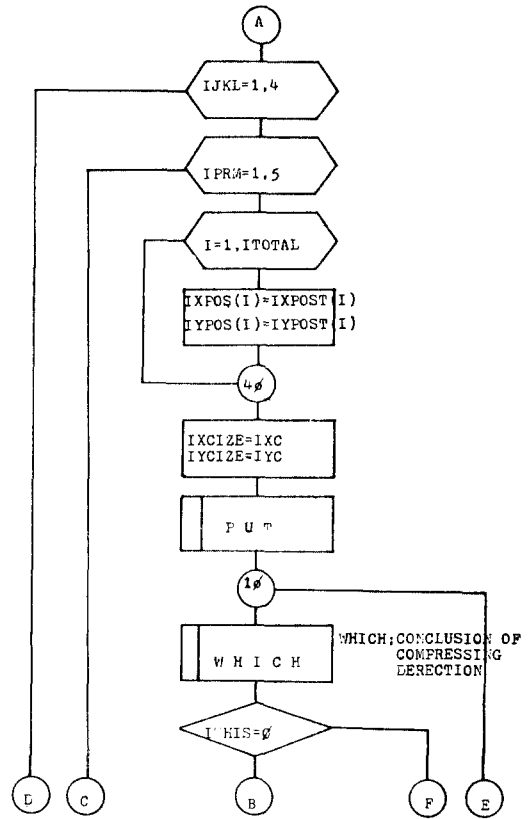


그림 2

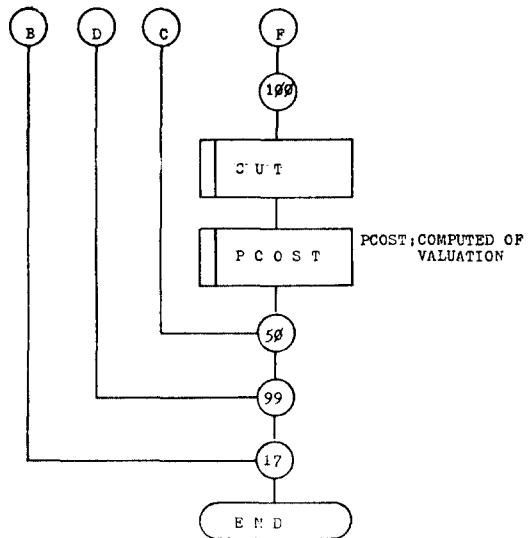


그림 3

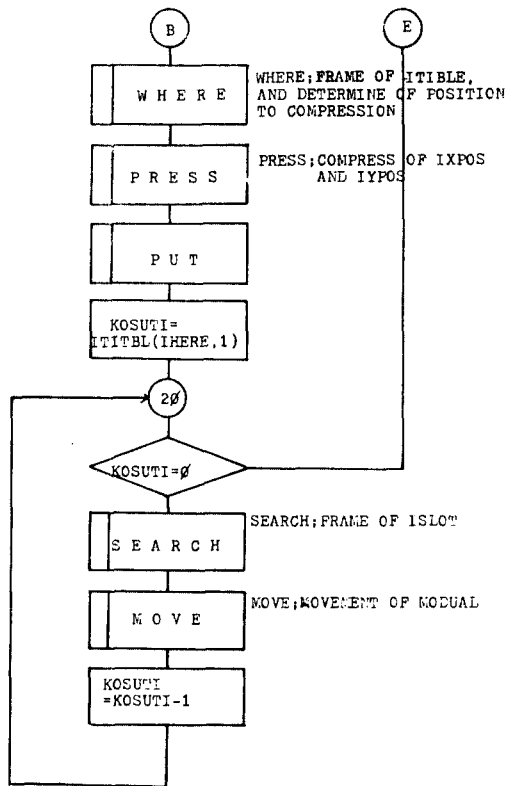


그림 4

参 考 文 献

- (1) B. Lattin, "VLSI design methodology," Proc. 16th D. A. Conf., p. 548, 1979.
- (2) 可兒, "LSI의 CAD," 信學誌, vol. 61, p. 710, 1979年 4月
- (3) M. Hanan, and J. M. Kurtzberg, "Placement techniques in chap. 5 in Design automation of digital systems: theory and techniques," vol. 1, Breuer M. A Edit, Englewood chiffs, New Jersey: Prentice-Hall, pp. 213-282, 1972.
- (4) J. M. Geyer, "Connection routing algorithm for printed circuit boards," IEEE Trans on CT, vol. 18, p. 95, Jan, 1971.
- (5) M. Hanan, P. Wolff, and B. J. Agule, "Some experimental results on placement techniques," Proc. 13th D. A. Conf., p. 214, 1976.
- (6) 鄭正和, "LSI의 Layout CAD," chap 3, 早稻田大學博士請求論文, pp. 99-100, 1980.
- (7) C. Y. Lee, "An algorithm for path connection and its application," IRE Trans. on Elect. Comp. p. 346, Sept. 1961.



朴 鎭 秀 (Jin Soo PARK) 正會員
 1948年 8月30日生
 1975年 2月: 漢陽大學校工科學電子工學科卒業
 1977年 2月: 漢陽大學校大學院電子通信工學科卒業 (工學碩士)
 1980年 9月~現在: 漢陽大學校大學院電子通信工學科 (博士過程)

1978年 3月~現在: 清州大學校電子工學科助教授