

Net-List로 부터 Schematic Diagram 자동 생성기의 배치 알고리즘

(Placement Algorithm for Automatic Generator of Schematic Diagram Using Net-List)

金 桓 石*, 李 天 熙**

(Hwan Seog Kim and Cheon Hee Yi)

要 約

본 논문은 도면설계를 자동으로 처리하기 위한 배치 알고리즘을 제안하였으며 신호선의 흐름에 따라 배치를 행하였다. 그러기 위하여 전체 도면을 각각의 모듈에 따라 subtree로 분할하였고 다시 tree는 x축과 y축의 상대적 좌표를 갖으면서 상위 level의 tree로 병합되는데 최종적으로 생성된 tree는 각 component 끼리 균형적인 공간을 갖게된다.

자동 도면설계를 위한 배치 알고리즘은 크게 Initiate, Divide, Merge, Placement의 4단계로 구성된다.

Abstract

In this paper, we developed placement algorithm for drafting logic schematics. The algorithm of this system is to base placement on signal flow as opposed to connectivity.

The key idea of this algorithm is that by dividing the schematic into simpler acyclic, uni-directional graphs, called tree. Then the components of tree and their interconnections have the characteristic that they can be laid out symmetrically without any crossovers. The interconnectivity between the trees are analyzed and a placement scheme is determined for them. Also, the trees are expanded into their respective components. This algorithm is divided into four main phases: Initiate, Divide, Merge, Placement.

I. 서 론

오늘날 반도체 산업의 지속적인 발달로 IC의 집적도가 날로 높아져 가는 추세이며 소량의 다품종 IC를 짧은 시간내에 설계가 요구되고 있는 실정이다. 따라서 거의 모든 IC설계에 컴퓨터를 이용하는 것이

보편화되었고 더불어 컴퓨터를 이용한 설계를 자동화하는 CAD도 전 설계과정에서 나날이 발전해 왔다!^[1]

그러나 이러한 DA (Design Automation) tool의 발달에도 불구하고 schematic design 만큼은 아직도 사용자로 하여금 심볼과 심볼사이의 연결관계를 직접 처리하고 있어 모든 IC설계의 기본인 schematic design에 많은 시간을 소비하고 있다. 본 논문은 이러한 schematic design에 있어서 각 Symbol을 자동으로 배치하기 위한 알고리즘을 제시하였으며 자동으로 생성되는 심볼들의 배치가 미적인 감각을 살려 전체 회로에 대한 신호의 흐름과 함수적 기능을 파악하

*正會員, **準會員, 淸州大學校 電子工學科
(Dept. of Elec. Eng., Chongju Univ.)

接受日字: 1989年 9月 30日

(※ 본 연구는 ETRI의 지원으로 이루어진 것임.)

도록 고려하였으며 사용자의 수작업으로 인한 실수로 회로설계시간이 길어지는 것을 방지할 수 있도록 하였다.

이러한 도면설계의 자동화는 다음과 같은 특징을 갖는다.^{[11][5]}

(1) 효율성증대

도면설계를 자동화함으로써 사용자가 schematic editor 상에서의 작업을 제거하여 전체설계시간을 단축하여 빠른 turn around를 갖음.

(2) 도면설계의 표준화

사용자 임의의 설계방식으로 설계도면의 함수적 기능 및 신호흐름 관계의 부정확한 이해를 방지하기 위해 컴퓨터에 입력된 표준화로 도면을 생성함으로써 설계도면의 통일된 서식화 가능.

(3) 재편집시 구조화된 도면 유지

구조화된 설계도면을 편집할 경우 전체적인 구조의 변화로 기능적 관계를 이해하기 힘든 상태이나 컴퓨터를 이용한 도면의 자동설계시에 전체도면을 항상 기능적 부분으로 구조화된 도면상태를 유지

(4) 선택적 화면출력

사용자가 도면을 한번 제작하면 그 형식을 바꾸기 어려우나 도면 설계자동기는 사용자의 요구에 따라 형식을 바꿀 수 있음

(5) 설계자동화증대

도면설계를 자동화함으로써 전체 설계과정을 자동화할 수 있음.

II. 알고리즘의 구성

논리 schematic을 자동생성하기 위해서는 PCB 나 LSI의 자동설계를 위한 알고리즘과 마찬가지로 배치와 배선문제가 그 기본이 된다. 배선문제는 그 간에 channel router가 많이 개선되어서 그것을 그대로 사용하더라도 별 문제가 없다. 그러나 논리 schematic을 자동생성하는데 있어서 배치문제는 PCB나 LSI와 같이 최적면적과 배선길이를 줄이는데 중점을 두는 것이 아니라 위상적인 신호흐름에 초점을 두고있기 때문에 PCB나 LSI의 배치기법을 그대로 사용할 수 없다.^[10]

논리 schematic의 중요한 기능의 하나는 디지털시스템의 기능을 명시하는 것인데 디지털시스템은 신호의 흐름으로 구성되어 있으며 이 신호에 따라 동작한다. 따라서 schematic의 흐름은 역으로 연결 길이, 전체연결길이, 굴곡수와 교차수가 최소가 되게 하고 대칭적으로 만들어야 한다.^[6]

신호의 흐름은 좌에서 우로, 밑에서 위 방향으로

하고 배치를 위해 schematic 평면을 동일한 크기의 단위블럭으로 나누고 소자와 연결선 사이에 적당한 간격을 유지하기 위하여 그리드 방법(grid method)을 사용하며 각 gate들의 배치는 단위 블럭에 의해 형성되는 그리드 좌표값에 따라서 결정된다.^[3,2]

그리드 구조에서 소자배치를 결정하기 위해서는 논리 schematic을 그래픽적인 회로망의 형태로 나타내야 하며 수학적으로 회로망은 source와 sink와 vertex의 매개수들로 구성된 방향성그래프(directed graph)로서 정의 되어야 한다.^[9]

본 연구에서 구성할 알고리즘의 주 목적은 되도록 선형설계로서, 상호의 흐름을 일반적으로 단일 방향으로 흐르며 갑작스런 방향의 변환이 없는 직선의 상태이다. 선형의 신호 흐름을 증진시키는 설계를 만들기 위한 divide and conquer 알고리즘을 사용하며 schematic을 acyclic이나 tree라 부르는 단일 방향성의 그래프로 분할한다.^[4] 도면배치 알고리즘(schematic placement algorithm)은 초기화(initiate), 분배(divide), 합병(merge), 배치(placement), 배선(routing)의 5단계로 나누어 진다.

1. 초기화(Initiate)

이 부분은 algorithm의 초기화 부분으로서 주어진 net-list로부터 심볼의 이름과 사용된 입출력단자, 신호선 이름, 상호관계를 유출해 내는 부분이다. 이러한 정보는 몇개의 자료구조로 저장함으로써 각 심볼간의 상호 연결관계를 알 수 있다. 본 논문에서 사용할 symbol에 관한 정보는 symbolic library에 내장된 SYM의 file에서 얻게되며 이 연구에선 단지 심볼간의 연결관계를 유출하면 된다.

신호선 이름에 대해서는 여러개의 분리된 신호선 중에서 같은 이름을 갖는 신호선에 대해서는 최종적으로 하나의 신호선으로 연결을 하는 방식으로 신호선을 처리했다. 이 부분에서 생성될 자료구조는 전체 schematic에서 사용되는 외부 입출력 단자에 대한 정보와 게이트들에 대한 정보를 주어진 net-list로부터 유출하는데 그에 대한 구조^[8]는 그림1, 그림2와 같다.

본 연구에서는 제약조건으로 combinational circuit에 한정으로 주어진 회로가 feedback이 있는지 없는지를 고려해야 한다. 1차적인 조사방법은 다음과 같다.

- 1) 외부 출력 port와 외부 입력 port의 연결관계
- 2) Signal net 탐색에서 같은 net에 있는 node의 재사용

만일 이와 같은 조건이 반복 될 경우는 sequential circuit으로 단정하고 실행이 종료된다.

File name,	primitive I/O num
1st primitive input port name,	signal name
2nd primitive input port name,	signal name
.	.
.	.
last primitive input port name,	signal name
1st primitive output port name,	signal name
.	.
.	.
last primitive output port name,	signal name

그림 1. 외부 입력단자에 대한 정보
 Fig. 1. Information of primitive I/O port.

Gate name	, I/O pin number
1st input pin name	, counting signal net name
2nd input pin name	, counting signal net name
"	,
"	,
last input pin name	, counting signal net name
out put pin name	, counting signal net name

그림 2. Gate에 대한 정보
 Fig. 2. Gate information.

2. Divide

Divide 단계에서는 이전 단계의 schematic에 관한 외부 입출력 단자에 대한 정보와 각 소자에 대한 정보를 이용해서 단일방향의 feedback이 없는 tree형식, 즉 acyclic 부분 그래프로 분할하며 기본 출력 신호에서부터 깊이우선 탐색법을 이용하여 schematic의 경로를 탐색한다. 이렇게 찾아낸 각각의 경로를 acyclic 그래프인 tree형식으로 나타내며 외부출력 port에서부터 거리를 나타내게 되며 이것은 나중에 각 gate들의 상대적 x좌표로 이용된다. 또한 tree구조를 이용함으로써 전체 schematic의 신호 흐름 관계를 자연스럽게 왼쪽에서 오른쪽의 단일 방향으로 재 정의가 가능하게 된다.

1) Tree로의 분할

Schematic을 tree로 분할하기위해서 initiate부분에서 정의된 외부출력선(Primitive-Output-Port)에서 주어진 signal name을 이용하게 된다. 이때 이 신호선 영역에 연결된 schematic의 gate는 각 tree에 해당하는 root의 tree성분이 정의하는 곳에서 하나의

seed로 동작하게 된다. 이 root에 연결된 gate에 입력된 순서로 하부 gate를 탐색하며 각각의 tree를 생성하면 된다. 그림3에서 tree 분할의 예를 나타냈다.

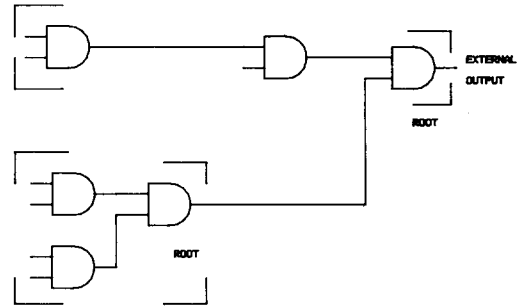


그림 3. Tree분할
 Fig. 3. Dividing of tree.

2) 상대적인 Gate의 좌표지정

이전 단계에서 각각의 gate들은 경로에 따라 tree로 구성된 상태이며 각 tree에 있는 해당 gate 들은 같은 tree내의 상대적인 수평좌표(X축)가 결정되어진 상태이다. 이 단계에서는 이와 같은 상대적 수평좌표와 각 gate들의 입력핀의 갯수에 따른 입력핀의 배치에 의해 부과된 게이트의 수평성분이 산출되므로 이것을 각 게이트들에 대한 Y좌표로 결정된다. 이때 고려해야 할 것이 balanced tree 기법인데 이것은 subtree를 먼저 적용하고 여기서 생성된 새로운 tree를 다시 상위단계의 tree와 결합하여 각각의 gate가 중첩됨이 없이 상대적인 위치에 배치하는 것이다. 이러한 배치에 사용되는 상대적인 X, Y축의 범위에 관한 정보를 다루기 위한 좌표구조가 필요하며 이것은 각각의 subtree에 대하여 생성되며 이것을 이용하여 각각의 subtree를 상대적 X, Y축에 따라 합병시킬 수 있는 것이다. 예로서 그림6을 고찰해 보면 각각의 X좌표에 대한 Y축의 범위에 대한 분석결과이며 그에 사용된 자료구조를 나타낸 것이다. 여기서 하나의 subtree는 (A, B, C)로 구성되며 또하나의 subtree를 (D)로 나타냈을 경우 이 두개의 subtree는 (E)와 병합되어 다시 상위 단계의 tree로 병합된다. 이때 X좌표 1에 있는 gate의 Y축 범위는 1에서 5이며 X축 2에 있는 gate는 2개이며 이들의 y 축 범위는 1에서 5이다.

이런 식으로 전개하여 그림4와 같이 나타낼 수 있다.

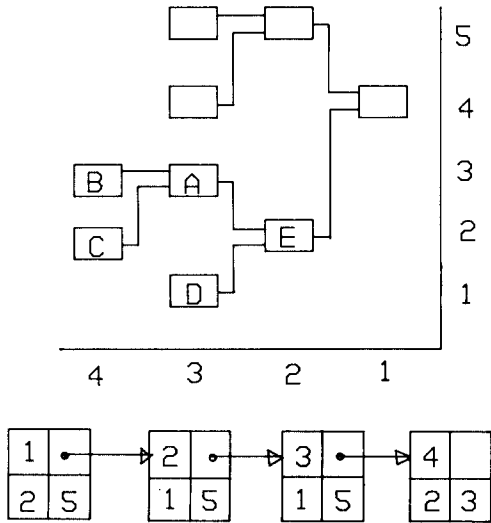


그림 4. 범위 데이터 구조
Fig. 4. Range data structure.

3) Subtree의 병합

범위 자료구조를 병합하는 방법은 다음과 같다. 예로 tree A와 tree B를 병합하기 위해서는 tree A의 첫번째 구성원과 tree B의 첫번째 구성원에서 최소 Y값을 구해 새로 생성되는 자료의 첫번째 Y 최소값에 놓고 새로 생성될 첫번째 Y 최대값에는 subtree A의 첫번째 최소 Y subtree B의 첫번째 Y의 최대값이 저장되며 다음단계 즉 X축 2의 구조에 Y 최대값은 off set=1을 더한 값이 저장되어야 한다. 부분 tree끼리의 병합된 결과는 다시 보다 상위의 level 성분과 다시 병합된다. 이때 사용될 root node로 gate의 위치는 병합된 출력성분에 따라 결정된다. 이러한 방법은 최종적으로 산출되는 tree가 balanced tree를 만족하기 위해서 있다.

3. 병합

이전 단계까지는 주로 하나의 tree내의 subtree에 대해서 고려하였으나 이 단계에서는 전체적인 각각의 tree의 상대적인 순서를 결정하며 tree를 이러한 순서에 의해서 공간배치를 행하는 것으로 문제의 간략화를 위해 각각의 tree를 하나의 모듈로 고려하였다. 이단계는 pass 1과 pass 2단계로 구분할 수 있다.

1) Pass 1—tree를 열에 할당

이 단계에서 tree는 신호선의 흐름에 따라 열의 왼쪽에서 오른쪽 방향으로 출력된다. 제일 좌측열은 입력신호에서 입력을 받는 tree로 구성되며 각각

의 열에 대한 tree는 이런 단계의 출력을 입력으로 취하는 tree들로 구성된다. 각 tree의 성분들은 root 성분에 대한 상대적인 X-좌표를 할당하며 root에서 leaf쪽으로 X-좌표값은 증가하게 된다. Tree를 다른 열에 포함시킴으로서 연결된 tree의 root에 대한 상대적인 X-좌표값을 계산할 수 있다. 마찬가지로 모든 schematic성분에 대한 상대적인 X-좌표를 구할수 있으며 X의 상대적 좌표값을 왼쪽에서 오른쪽으로 증가하게끔 각 tree를 열에 할당한다. 이렇게하여 schematic의 왼쪽에서 tree의 root에 대한 상대적 X값을 결정할 수 있으며 root X-좌표값은 다음 2가지 중 하나의 최대값으로 결정된다.

가) Tree의 크기

나) Tree의 leaf를 나타내는 list값으로부터 최대 X-좌표값과 leaf의 입력핀의 오른쪽에 연결되어 있는 root의 X-좌표값

2) Pass 2 - 각열내에 tree를 배열

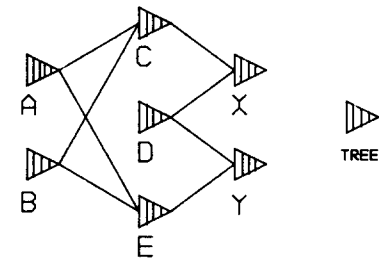
각열내에 있는 tree를 수직적으로 배열하기 위해 tree간의 연결도에 따라 heuristic하게 사용되는데 이때의 연결도는 공유되는 입출력의 수와 깊이 그리고 연결입력 성분핀의 순서에 기초한다. 이 연결도를 주어진 열에서 모든 가능한 tree쌍을 만들어 이들을 가장 연결도가 높은곳에서 가장 낮은 곳으로 bubble sort를 행하며 입력 연결을 주어진 tree쌍에서 공유되는 입력의 깊이와 수를 기준해서 나타낸다.

이 과정에서 사용되는 depth factor table [표1]은 교차깊이 level을 연결값으로 산출해내는데 (그림 5) 도움이 된다.

Depth가 1인 경우의 값은 1이고, 2인 경우에 값은 1/2, 4인 경우에는 1/4이 되며 연결수가 2개씩 증가할 때마다 값은 1/2씩 감소한다. 표1은 potential 교차수가 깊이 level이 거의 지수적으로 증가(그림6)하는 곳에 사용하며 그림6은 입력 연결도를 분석한 그림이다.

표 1. 깊이 요소 표
Table 1. Depth factor table.

Depth	VALUE	Depth	VALUE
1	1	6	.125
2	.5	7	.0937
3	.37	8	.0625
4	.25	9	.0465
5	.187	10	.30125

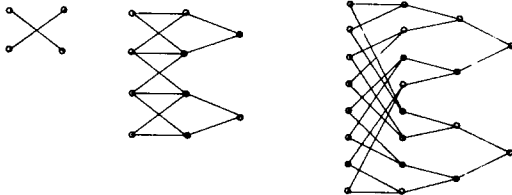


INPUT CONNECTIVITY BETWEEN TREE A & B

PATH	DEPTH	INPUT CONNECTIVITY VALUE
X-D-Y	2	1/2
X-C-A-E-Y	4	1/4
X-C-B-E-Y	4	1/4

TOTAL 1

그림 5. 입력 연결값 산출
Fig. 5. Measuring input connectivity.



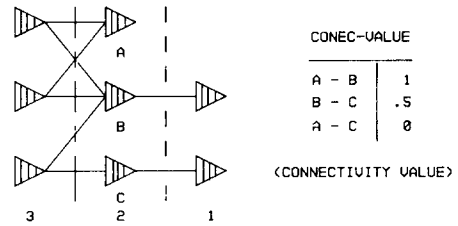
SEGMENT LENGTH (2N)	2	4	6
NUMBER OF PATHWAYS (2N)	2	4	8

그림 6. 입력연결도 분석
Fig. 6. Input connectivity analysis.

3) Ordering 과정

연결도는 주어진 열의 모든 가능한 tree쌍에서 결정되며 이러한 쌍은 분류되어 CONEC-VALUE 이라 하는 connectivity-value 자료구조에 저장되는데 이 정보는 tree가 함께 그룹되고 열내에서 배치될 때의 순서를 결정하기 위해 사용된다. CONEC-VALUE 은 높은 연결값을 가지며 entry에 대응하여 SELECT 라 하는 또다른 자료구조가 만들어 진다.

이 자료구조는 CONEC-VALUE, entry의 tree 2개를 포함한 2개의 그룹에 대한 pointer를 가지고 있다. CONEC-VALUE의 ordering entry는 SELECT자료



GROUPS : A, B, C

(TAKE HIGHEST CONEC-VALUE, ENTRY(A-B) & MAKE A SELECT ENTRY)

SELECT

A: B: NO PREFERENCE OVER ORIENTATION
=> MERGE GROUP A ON TOP OF GROUP B

GROUPS : A-B, C

(TAKE HIGHEST CONEC-VALUE ENTRY (B-C) & MAKE A SELECT ENTRY)

SELECT

A-B : C -A-B OVER C IS PREFERRED
SINCE B'S OUTPUT IS ABOVE C'S IN COLUMN 1

=> MERGE GROUP A-B ON TOP OF GROUP C

=> THE ORDER OF COLUMN 2 IS A OVER B OVER C

그림 7. CONEC-VALUE, SELECT에 의한 tree의 ordering

Fig. 7. Ordering trees in a column using CONEC-VALUE, SELECT.

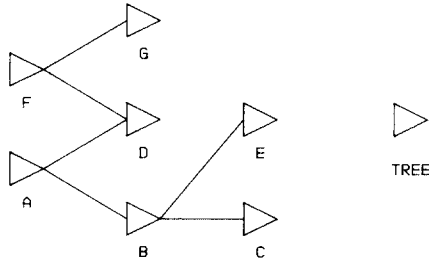
구조에 의해 그룹으로 merge된 ordering을 결정해준다.

4. 배치 (Placement)

이 단계에서는 이전 단계에서 순서화된 tree의 column을 가지고 단위 좌표공간의 tree에 대해 상대적 배치를 결정하는 단계이다. 여기서 사용되어지는 자료는 병합단계에서 생성된 range-structure가 사용되어지며 이 자료는 tree가 배치되면서 갱신 되어진다. 여기서 두가지 단계로 분할 되는데 그 첫번째가 scan이며 다음 단계는 update이며 이에 대한 algorithm은 그림8에 나타낸다.

pass1에서는 입력쪽에서 출력방향으로 scan되며 marking된다. Scan되는 순서는 최하단 레벨 tree의 최 좌측열부터 연결된 tree가 없을 때까지 탐색을 한다. 예로써 그림8에서 보는 바와 같이 pass1에서 A, B, C순서로 scan된다.

pass 2에선 pass 1에서 표시된 tree의 출력에서 입력방향을 추적하면서 만나는 module을 배치하게 되는 것이다. 예로서 그림8에서 먼저 C와 B가 배치되고 다시 module E를 scan하여 E를 배치하고 최종적으로



```

SS 1 : SCAN A-B-C
PASS 2 : PLACE C
        PLACE B
        UPDATE : SCAN E, PLACE E
        PLACE A
        UPDATE : SCAN D

SS 1 : SCAN F-D
PASS 2 : PLACE D
        PLACE F
        UPDATE : SCAN G, PLACE G
  
```

그림 8. 배치
Fig. 8. Placement.

로 A를 배치한다. 이때 A와 D의 연결 관계로 D를 scan하였으나 D가 F에 종속된 module이므로 F-D tree부분에서 D의 배치를 하게된다.

III. 실험 및 고찰

본 논문에서 제시한 알고리즘에 의한 생성기로 그림9와 그림10을 예로 보였으며 그림9에 대한 net-list는 워낙 긴 data여서 생략하고 그림10에 대한 net-list만 나타내었다. 아직 화면출력을 plotter로 출력을 하진 못해서 명확하게 출력을 보진 못하지만 주어진 net-list로 모든 심볼을 배치하고 신호선에 맞게 배선된 것을 알 수 있었다. 배선을 위해서 사용한 방법은 일반적으로 널리 사용되고 있는 channel router를 사용하였기 때문에 신호선의 연결과정에서 많은 굴곡수를 보이고 있으나 현재 전용 배선방식을 연구중이므로 조만간 자동 도면생성기의 배치문제뿐 아니라 배선 문제도 해결될 것이다.

IV. 결 론

Logic schematic을 자동생성하는데 Initiate, Divide, Maerge, Placement, Router의 5단계로 나누어 실행

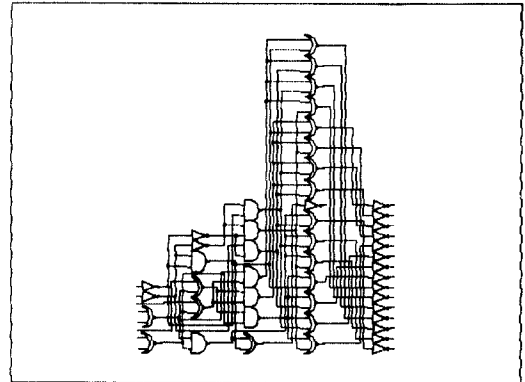


그림 9. 자동 도면 생성기에 의한 출력 예 1
Fig. 9. Example 1 of output by schematic generator.

```

IN, 26,
OUT, 14,
NAND3, 20,
4, 0,
IN1, 11,
IN2, 17,
IN3, 14,
OUT, 20,
NAND3, 21,
4, 0,
IN1, 11,
IN2, 18,
IN3, 14,
OUT, 21,
NAND3, 22,
4, 0,
IN1, 16,
IN2, 17,
IN3, 14,
OUT, 22,
NAND3, 23,
4, 0,
IN1, 16,
IN2, 18,
IN3, 14,
OUT, 23,
END, END,
p112,
A, 24,
B, 25,
C, 26,
D0, 20,
D1, 21,
D2, 22,
D3, 23,
INV, 15,
2, 0,
IN, 24,
OUT, 11,
INV, 16,
2, 0,
IN, 11,
OUT, 16,
INV, 17,
2, 0,
IN, 25,
OUT, 17,
INV, 18,
2, 0,
IN, 11,
OUT, 11,
INV, 19,
2, 0,
  
```

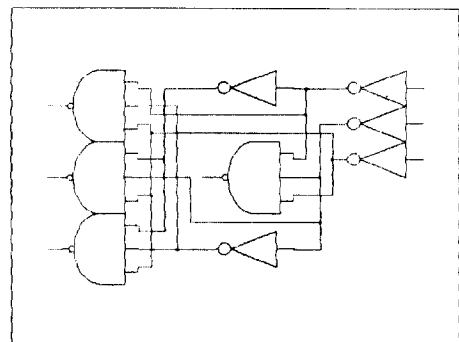


그림 10. 자동도면 생성기에 의한 출력 예 2
Fig. 10. Example 2 of output by schematic generator.

하였으며 복잡도를 줄이기 위하여 Divide and Conquer 방법을 사용하였고 분할의 단위는 tree로 하였다. tree는 한개의 신호원 소자로부터 새로운 신호 set가 어떻게 상호 작용하는지를 나타내고 있기 때문에 논리 분할점이 된다.

Schematic을 tree로 그룹지음으로써 설계과정에서 고려하여야 할 상호작용의 수를 감소시켰고 이 감소된 수를 가지고 heuristic기법을 적용하여 tree는 실제소자로 전개하였다. 배치단계에서는 각 level 상호간의 연결도 및 wire 꺾어짐의 갯수등을 고려하여 각 symbol들의 상대적인 y좌표를 구하여 symbol들을 배치하였다.

배치단계에서는 signal flow는 입력단에서부터 출력단까지 일괄성을 유지하는 것이 사용자가 회로도를 이해하기 쉬우므로 각 symbol들의 연결도를 방향성 그래프로 표시하여 이 좌표를 구하였고 배치프로그램에 대한 새로운 알고리즘을 제시하였다.

현재는 EDAS-P로부터 net-list를 받아 gate-level digital circuit 범위내에서 구현하였으나 앞으로는 interface만 만듦으로써 VHDL이나 silicon compiler로부터 받은 net-list를 사용하여 schematic diagram을 만드는 것도 가능하다.

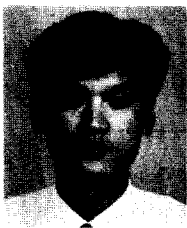
디지털시스템의 CAD분야는 급속히 성장하고 있으므로 논리 schematic을 컴퓨터를 사용하여 자동으로 설계하는 연구도 점점 늘어날 전망이다.

參 考 文 獻

- [1] T. Ohtsuki, "Layout design and verification," North Holland, 1986.
- [2] N. Kukreja, "AGOS: Automatic generation of schematic layout," M. Tech. Project, EE Dept, Indian Inst, Technology Delhi, 1983.

- [3] A. Srivastava, "SCHELAG: schematic layout generation package," M. Tech report, EE Dept. Indian Inst. Technology Delhi, 1984.
- [4] M.L. Ahlstrom, G.D. Hadden and G.R. Stroick, "HAL: A heuristic approach to schematic generation," ICCAD, pp. 83-86, 1984.
- [5] Anjali Arya, Anshul Kumar, V.V. Swaminathan, Amit Misra, "Automatic generation of digital system schematic diagram," DAC, pp. 388-395, 1985.
- [6] Robert K. Chun, Keh-jeng chang, Lawrence P. Mcnamee, "VISION: VHDL induced schematic imaging on net-list," 24th, ACM/IEEE Design Automation Conference, pp. 436-441, 1987.
- [7] G. Zimmermann, "The MIMOLA design system, A computer aided processor pp. 53-58, 1979.
- [8] B.T. Preas, C.W. Gwyn, "Methods for hierarchical automatic layout of custom LSI integrated circuit," Proc. 15th Design Automation Conference, pp. 206-212, 1978.
- [9] Ashok Kumar, et al, "Automatic generation of digital systems," IEEE Design and Test, Feb, pp. 58-65, 1986.
- [10] Prease, B.T., "Placement and routing algorithms for hierarchical innegrated circuit layout," Ph.D. dissertation, Stanford University, 1979.
- [11] Carver Mead, Lyan Conway, "Introduction to VLSI," Addison 1980."

著 者 紹 介



金 桓 石 (準會員)

1963年 3月 1日生. 1988년 2월
청주대학교 전자공학과 졸업. 현
재 청주대학교 전자공학과 석사과
정. 주관심분야는 VLSI설계와
CAD TOOL개발 등임.

李 天 熙 (正會員) 第26卷 第8號 參照

현재 청주대학교 전자공학과
교수.