

論文2002-39SD-3-7

1-Steiner 트리 알고리즘을 응용한 시간 지향 배선 방법 (Timing-Driven Routing Method by Applying the 1-Steiner Tree Algorithm)

沈 昊 , 林 鐘 錫

(Ho, Shim and Chong Suck, Rim)

요 약

본 논문에서는 1-Steiner 휴리스틱 알고리즘을 응용하여 단일 소스 넷과 다중 소스 넷을 배선하는 두 가지 시간 지향(timing-driven) 배선 방법을 제안한다. 이 방법은 1-Steiner 휴리스틱 알고리즘의 계산값(cost)을 지연시간으로 수정한 것으로 이 방법의 특징은 모든 터미널이 임계터미널인 경우와 또 임계터미널이 부분적으로 존재하는 경우의 단일 소스 넷과 다중 소스 넷을 배선하는 데 동시 적용할 수 있다는 점이다. 실험결과 단일 소스를 배선하는 알고리즘은 기존의 SERT와 SERT-C에 비해 지연시간이 각각 평균 2.1%, 10.6% 감소하는 성능을 보였다. 그리고 다중 소스를 배선하는 알고리즘은 기존의 MCMD A-tree 알고리즘과 비교했을 때 모든 소스, 터미널 쌍이 임계쌍(critical pair)일 경우는 최대 지연 시간이 평균 2.7% 증가했지만 부분적인 임계쌍이 존재할 때는 최대 지연 시간이 평균 1.4% 감소하는 유사한 결과를 도출한다.

Abstract

In this paper, we propose two timing-driven routing algorithms for single-source net and multi-source net as applications of 1-Steiner heuristic algorithm. Using the method of substituting the cost of 1-Steiner heuristic algorithms with interconnection delay, our routing algorithms can route both single-source net and multi-source net which have all critical source-terminal pairs or one critical pair efficiently. Our single-source net routing algorithm reduced the average maximum interconnection delay by up to 2.1 % as compared with previous single-source routing algorithm, SERT, and 10.6 % as compared with SERT-C. and Our multi-source net routing algorithm increased the average maximum interconnection delay by up to 2.7 % as compared with MCMD A-tree, but outperforms it by up to average 1.4 % when the signal net has only subset of critical node pairs.

I. 서 론

Deep-submicron 단위의 VLSI 회로 공정기술이 발달

* 正會員, 三星電子 半導體 研究所 메모리 CAE 팀
(Samsung Electronics Semiconductor R&D Center
Memory CAE Team)

** 正會員, 西江大學校 컴퓨터學科
(Dept. of Computer Science and Engineering, Sogang
University)

接受日字:2001年8月8日, 수정완료일:2002年1月14日

함에 따라 회로의 연결 지연시간(interconnection delay)이 회로의 전반적인 성능을 결정하는 주 요소가 되었다. 이에 따라 배선 최적화 방법에 대한 연구들이 활발하게 진행되어 왔다.

배선 최적화에 대한 연구는 기술의 발전에 맞추어 각기 다른 각도에서 진행되어 왔다. 와이어의 저항이 전체 회로에 미치는 영향이 미비했을 때는 주로 MRST(Minimum Rectilinear Steiner Tree) 문제를 배선에 응용하기 위한 연구들이 수행되었다^[1,2].

그러나 점차 공정기술이 deep-submicron 단계로 접어들어 따라 와이어의 저항이 넷의 연결 지연시간

(interconnection delay)에 미치는 영향을 무시하지 못하게 되었다. 따라서 배선의 목적이 와이어의 정전용량 최소화에서 와이어 연결형태 최적화(topology optimization)로 옮겨갔다. 이에 따라 와이어의 연결형태를 고려한 A-Tree^[3], SERT^[4], BB-SORT^[5], MVERT^[6] 등의 배선 알고리즘들이 제안되었다. A-Tree^[3] 알고리즘은 소스에서 각 터미널까지의 최단 길이를 갖는 트리를 구성하는 알고리즘이다. 이 알고리즘을 배선에 이용했을 경우 배선에 사용되는 전체 와이어의 길이가 최소화하면서 동시에 지연시간을 단축할 수 있다. SERT^[4]는 Elmore 지연시간(Elmore delay)^[7]의 충실도^[8]를 바탕으로 네트의 최대 지연시간을 최소화 해주는 알고리즘이고, SERT-C^[4]는 네트에 임계터미널이 하나 존재할 경우에 그 임계 터미널까지 지연시간을 최소화 해주는 알고리즘이다. 이 알고리즘들은 빠른 실행시간 안에 기존의 MST나 AHK^[9] 알고리즘에 비해 효과적인 배선을 수행하지만 그리디(greedy) 알고리즘이므로 네트에 속한 터미널의 수가 많아지는 경우에는 최적의 결과와 차이를 보인다. BB-SORT^[5]와 BB-SORT-O^[5]는 SERT를 보완하는 알고리즘으로써 제안되었다. 그리디 알고리즘의 SERT와 달리 Branch-and-Bound 형식의 다이내믹(dynamic) 알고리즘이기 때문에 SERT에 비하여 지연시간과 배선에 사용되는 와이어의 길이 면에서 보다 효과적인 배선을 수행한다. 하지만 다이내믹 알고리즘이므로 배선할 네트에 속한 터미널의 수가 많아질 경우 실행시간이 지나치게 길어진다. 또 MVERT^[6]는 각 터미널에 도착 요구 시간이 명시되었을 경우 이 도착 요구 시간을 만족시키는 한에서 배선에 사용되는 와이어의 길이를 최소화하는 Non-Hanan 점을 찾아 배선을 하는 알고리즘이다. 그러나 먼저 Hanan 점을 이용한 배선된 네트가 입력으로 주어지기 때문에 이 입력되는 배선트리의 성능에 그 결과가 제약을 받는다. 그리고 최근에 배선의 최적화를 위하여 많이 연구되는 것이 버퍼 삽입과 와이어 너비조절(wire sizing)을 통한 지연시간 단축방법이 있으나 본 연구의 범위를 벗어나므로 생략한다.

또 최근 다중 소스를 가진 네트의 배선에 대한 연구도 병행되고 있다. Cong 등^[10]은 다중 소스 네트 배선 알고리즘으로서 트리의 각 소스와 각 터미널간의 길이 중 가장 긴 것을 그 트리의 지름(Diameter)이라고 정의하고 이 지름을 최소화하여 각 소스와 각 터미널 사이의 최대 지연시간을 최소화하는 MD A-Tree와 네트의

임계 소스, 터미널 쌍(pair)의 최대 지연시간을 최소화 하는 MCMD A-Tree를 제안하였다. 그리고 Lillis 등^[11]은 다중 소스 네트의 성능을 각 소스, 터미널 쌍간의 지연 시간들 중에서 최대 지연시간이라 정의하고 그 최대 지연시간을 $O(n)$ 시간에 측정하는 알고리즘과 다중 소스 네트에 쌍방향 버퍼(repeater)를 삽입하여 배선의 성능을 높이는 다이내믹 알고리즘을 제안했다. 또 Cong 등^[12]은 와이어 너비조절을 통해 다중 소스 배선의 성능을 향상시키는 알고리즘을 제안하였다. 그러나 [12,11]에서 제안된 알고리즘들은 네트의 연결형태를 구성해 주는 것은 아니고 미리 배선된 트리가 입력으로 주어졌을 때 그 트리의 성능을 향상시키는 후처리 배선 알고리즘의 성격을 띤다.

본 논문에서는 1-Steiner 휴리스틱 알고리즘을 응용하여 단일 소스 네트를 배선하는 STD(Single-source Timing Driven) 1-Steiner 알고리즘과 다중 소스 네트를 배선하는 MTD(Multi-source Timing Driven) 1-Steiner 알고리즘을 제안한다. 두 알고리즘은 기존의 1-Steiner 휴리스틱 알고리즘의 계산값(cost)을 배선트리의 길이대신 배선트리의 지연시간으로 수정한 시간 지향형 알고리즘들이다. 본 논문은 다음과 같이 구성된다. 먼저 제 2장에서는 제안된 배선 방법의 기본이 되는 기본정리(preliminary)를 기술하고, 제 3장에서는 STD 1-Steiner 알고리즘, 4장에서는 MTD 1-Steiner 알고리즘을 각각 기술한다. 제 5장에서는 제안한 배선 방법들과 기존의 배선 방법들과의 성능을 비교한 실험 결과를 제시한 후 마지막으로 제 6장에서 결론을 내린다.

II. 기본 정리

1. Elmore 지연시간 모델

Elmore 지연시간 모델^[7]이 제시된 이후 지금까지 Elmore 지연시간 모델은 비교적 계산이 간단하고 그 결과가 충실하기 때문에 배선된 네트의 연결 지연시간 측정을 위해 흔히 사용되어 왔다. 단위 길이 저항이 r_e , 단위 길이 정전용량이 c_e 이고 길이가 l 인 와이어 세그먼트의 Elmore 지연시간 D_{wire} 은 다음과 같다.

$$D_{wire} = r_0 \left(\frac{1}{2} c_0 + C_L \right)$$

여기서 r_0 는 $r_e l$, c_0 는 $c_e l$ 이고 C_L 은 와이어에 직접

연결된 다운 스트림(down stream) 정전용량이다.

다음으로 소스 p_0 를 가진 네트 $P = \{p_0, p_1, \dots, p_n\}$ 를 배선한 트리 $T(P)$ 가 주어졌을 때 소스 p_0 의 구동 저항 (driver resistance)를 R_d , $T(P)$ 의 노드 v 와 v 의 부모를 연결하는 간선을 e_v 라 하고, e_v 의 저항과 정전용량을 각각 r_{e_v} 와 c_{e_v} 라고 하자. 그리고 v 를 루트로 하는 부속트리에 속한 터미널과 간선의 정전용량의 합을 C_v 라 할 때 소스 p_0 에서부터 터미널 p_i 까지 Elmore 지연시간 ED_i 는 다음과 같이 계산된다^[13].

$$ED_i = R_d C_{p_0} + \sum_{e_v \in \text{path}(p_0, p_i)} r_{e_v} (\frac{1}{2} c_{e_v} + C_v)$$

2. Iterated 1-Steiner 휴리스틱 알고리즘

본 절에서는 본 논문에서 제안하는 배선방법의 바탕이 되는 Iterated 1-Steiner 휴리스틱 알고리즘을 간략하게 기술한다. VLSI 회로 디자인에서 MRST 문제는 개략 배선 등의 응용으로서 많이 연구되었다. 하지만 MRST 문제는 NP-complete이므로^[14] 기존의 MST를 이용하여 MRST를 구하는 휴리스틱 알고리즘들이 제안되었다^[1,2]. 그 중에서 1-Steiner 휴리스틱 알고리즘^[1]은 Hanan의 원리^[15]를 근간으로 하여 다항 시간 (polynomial time)에 MRST를 구하는 매우 효과적인 알고리즘이다. 평면상에 점들의 집합 P 가 주어졌을 때 Iterated 1-Steiner 휴리스틱 알고리즘은 반복적인 스타이너 점들의 삽입을 통해 P 의 MRST를 구하는 그리디(greedy) 알고리즘이다. 여기서 스타이너 점은 P 의 Hanan 그래프의 정점 중 P 에 속하지 않은 정점이다. Iterated 1-Steiner 휴리스틱 알고리즘을 개략 기술하면 다음과 같다. 먼저 1-Steiner 점은 다음과 같이 정의된다.

- 1-Steiner 점: 맨하탄 평면에 주어진 점들의 집합 $P = \{p_1, p_2, \dots, p_n\}$ 와 P 의 MST, $MST(P)$ 의 길이를 $L_{MST}(P)$ 라 할 때, P 의 Hanan 그래프의 정점 x 중에서 아래의 조건 (1),(2),(3) 모두를 만족하는 점 x .
 - (1) $x \notin P$, (2) $L_{MST}(P) > L_{MST}(P \cup \{x\})$,
 - (3) $L_{MST}(P \cup \{x\})$ 가 최소

또 특별히 Hanan 그래프의 정점 중 $x \notin P$ 인 모든

정점을 P 의 1-Steiner 후보점(candidate 1-Steiner point)이라고 언급한다. 1-Steiner 알고리즘은 맨하탄 평면상에서 점들의 집합 P 와 현재 공집합인 스타이너 점의 집합 S 를 가지고 시작한다. 먼저 P 의 1-Steiner 후보점 중에서 1-Steiner 점인 x 를 찾아 $S \leftarrow S \cup \{x\}$, $P \leftarrow P \cup S$ 로 갱신한다. 다시 갱신된 P 의 1-Steiner 점을 찾아 위와 같은 과정을 적용한다. 이러한 단계가 계속 되면 점차 $L_{MST}(P)$ 가 줄어들게 된다. 그리고 더 이상 1-Steiner 점을 찾지 못하면 프로그램을 마친다. 여기서 한 가지 언급할 것은 최소의 S 를 구하기 위하여 각 단계마다 $MST(P \cup S)$ 중 degree ≤ 2 인 스타이너 점들 S 에서 제거한다. 그림 1에 Iterated 1-Steiner 알고리즘을 보인다.

Iterated 1-Steiner Heuristic Algorithm
Input : A point set $P = \{p_1, p_2, \dots, p_n\}$
Output : Rectilinear Steiner Tree of P
<pre> S ← ∅ while 1-Steiner point x of P exists do S ← S ∪ {x} Remove degree ≤ 2 steiner points in MST(P ∪ S) from S P ← P ∪ S end while return MST(P) </pre>

그림 1. Iterated 1-Steiner Heuristic 알고리즘.
Fig. 1. Iterated 1-Steiner Heuristic Algorithm.

III. 단일 소스 네트의 배선 알고리즘

1. 문제 정의

일반적으로 단일 소스 네트 배선의 목적은 소스에서 임계 터미널(critical sink)들까지의 지연시간을 최소화 하는 것이다. 만약 네트의 모든 터미널들이 임계 터미널이라면 소스에서 각 터미널까지의 지연시간 중 최대 지연시간을 최소화하는 것이 배선의 목적이 된다.

p_0 를 소스로 가지는 단일 소스 네트 $P = \{p_0, p_1, \dots, p_n\}$ 가 주어졌을 때 배선을 하기 전 시간 분석

(timing analysis)을 통해 각 터미널 $p_i(i \neq 0)$ 에 가중치 W_i 를 연계시킬 수 있다. W_i 는 터미널 p_i 의 지연시간 임계도(criticality)를 나타내며 터미널의 도착 요구 시간이 빠를수록 큰 값을 가진다. 또 소스에서 터미널 p_i 까지 지연시간을 d_i 라고 한다면 $W_i d_i$ 는 각 터미널의 가중 지연시간(weighted delay)이 된다. 배선은 최대 가중 시간을 최소화하는 것과 배선길이를 줄이는 것을 목적으로 한다. 배선의 목적을 정리하면 다음과 같다.

- (1) $\max \{ W_j d_j \}$ 를 최소화하기($p_j \in P, j \neq 0$).
- (2) 전체 배선에 사용되는 와이어의 길이를 최소화하기.

본 논문에서 제안하는 STD 1-Steiner 알고리즘은 (2)보다 (1)에 더 우선 순위를 둔다.

가중치는 $[0, \infty]$ 사이에서 결정될 수 있다. 그러나 본 논문에서는 프로그램의 간결성을 위해 각 터미널에 가중치를 1 또는 0으로 할당한다. 즉 지연시간 임계도가 높은 터미널의 가중치는 1이고 그렇지 않은 터미널의 가중치는 0으로 한다.

2. STD(Single-source Timing Driven) 1-Steiner 알고리즘

STD 1-Steiner 알고리즘은 본 논문에서 제안하는 성능 지향형 단일 소스 배선 알고리즘이다. STD 1-Steiner 알고리즘은 반복적인 스타이너 점의 삽입을 통해 트리를 갱신해 나간다는 점에서 1-Steiner 휴리스틱 알고리즘과 유사하다. 그러나 1-Steiner 휴리스틱 알고리즘의 목적은 주어진 네트워크의 MRST를 구하는 것이지만 STD 1-Steiner 알고리즘의 목적은 트리의 최대 가중 지연시간을 최소화하는 것이다. 본 절에서는 STD 1-Steiner 알고리즘에 대하여 기술한다. 먼저 STD 1-Steiner 알고리즘의 효율적인 기술을 위하여 SM(Single-source Modified) 1-Steiner 점을 정의한다.

- SM 1-Steiner점 : p_0 를 소스로 가지는 단일 소스 네트워크 $P = \{ p_0, p_1, \dots, p_n \}$ 를 배선한 트리 $T(P)$ 가 주어지고 $mwd(T(P))$ 를 $T(P)$ 의 최대 가중 지연시간(maximum weighted delay)이라 할 때 P 의 Hanan 그래프 정점 z 중 아래의 조건 (1),(2),(3)을 동시에 만족하는 스타이너 점 z .
 - (1) $z \notin P$, (2) $mwd(T(P)) > mwd(T(P \cup \{z\}))$,
 - (3) $mwd(T(P \cup \{z\}))$ 가 최소

SM 1-Steiner 점을 이용하여 STD 1-Steiner 알고리즘의 수행과정을 기술하면 다음과 같다. 먼저 주어진 네트워크 P 의 배선트리 $T(P)$ 를 구성한다. 처음 구성되는 $T(P)$ 는 어떠한 구성형태를 이루어도 상관없다. 그러나 본 연구에서는 알고리즘의 성능과 수행시간 및 배선의 길이 등을 고려하여 초기 $T(P)$ 를 P 의 MST로 놓는다. 다음 P 의 SM 1-Steiner 점 z 를 구한다. 그리고 z 에 의하여 새롭게 구성된 트리를 $T(P \cup \{z\})$ 라 한다면 $T(P \cup \{z\})$ 는 SM 1-Steiner 점의 정의에 의하여 $T(P)$ 보다 최대 가중 지연시간이 작다. 만약 이때 z 가 존재하지 않으면 알고리즘을 마친다. 그렇지 않다면 $T(P) \leftarrow T(P \cup \{z\}), P \leftarrow P \cup \{z\}$ 로 갱신하고 다시 P 의 SM 1-Steiner 점을 찾는다. 이러한 과정을 더 이상 P 의 SM 1-Steiner 점이 존재하지 않을 때까지 반복 수행한다. 이렇게 STD 1-Steiner 알고리즘의 특징은 반복적으로 SM 1-Steiner 점을 찾아 트리의 최대 가중 지연 시간을 전 단계보다 줄여 간다는 것이다. 이것은 1-Steiner 휴리스틱 알고리즘이 현재 네트워크 P 의 원소와 1-Steiner 점의 MST를 통해 전단계 보다 트리의 길이를 줄여 가는 과정과 성격이 같다. 그러나 STD 1-Steiner 알고리즘은 1-Steiner 알고리즘에서 사용하는 선형 시간 MST 갱신(linear-time MST update) 알고리즘[13]을 조금 변형하여 효과적인 시간 지향형 배선을 수행한다.

3. 선형 시간 MST를 응용한 트리 발생

선형 시간 MST 갱신 알고리즘은 $MST(P)$ 가 존재할 때 새롭게 추가된 점 x 와 P 의 MST, $MST(P \cup \{x\})$ 를 $O(n)$ 시간에 구하는 알고리즘이다. 이 알고리즘을 사용함으로써 1-Steiner 휴리스틱 알고리즘은 현재 P 의 1-Steiner 점을 $O(n^3)$ 에 구할 수 있다. 선형시간 MST 갱신 알고리즘은 $O(n)$ 시간에 P 에 속하는 스타이너 점 x 의 잠재적인 이웃 정점들(potential neighbor points)을 계산한 후 이 이웃 정점들과 x 를 연결하여 사이클이 발생하면 사이클을 이루는 간선들 중 가장 긴 간선을 제거하는 과정을 거쳐 최종 $MST(P \cup \{x\})$ 를 얻는다. 1-Steiner 휴리스틱 알고리즘에서 사용하는 선형시간 MST 갱신 알고리즘의 자세한 설명은 [13]에 나온다. 이와 달리 STD 1-Steiner 알고리즘은 발생한 사이클을 이루는 간선들을 하나씩 제거해 보면서 그 중 트리의 최대 가중 지연시간을 최소화하는 간선을 찾아 사이클에서 제거한

다. 먼저 x 와 이웃 정점들을 연결한 간선들을 최대한 활용하기 위해 이웃 정점들을 x 와의 거리에 따라 오름 차순으로 정렬한다. 그리고 정렬된 순서대로 x 와 연결하여 사이클이 발생하면 트리의 최대 가중 지연 시간을 최소화하는 간선을 찾아 사이클에서 제거하면서 트리를 갱신해 나간다.

그림 2는 기존 MST가 존재할 때 새로운 스타이너 점 x 를 포함하는 새로운 MST를 구하는 선형시간 MST 갱신 알고리즘의 예제이다. 그리고 그림 3은 STD 1-Steiner 알고리즘이 트리의 최대 가중 지연 시간을 최소화하는 간선을 제거하여 트리를 발생시키는 예제이다. 다음 그림 4에 STD 1-Steiner 알고리즘을 보인다.

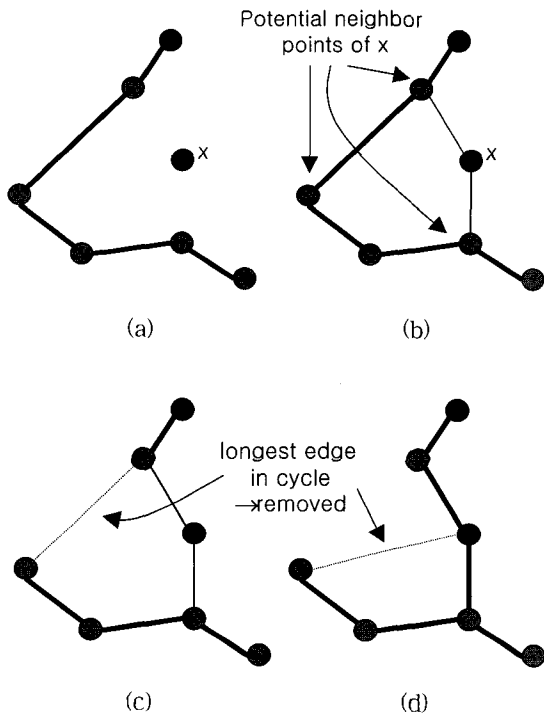


그림 2. (a) MST와 스타이너 점 x . (b) x 와 x 의 잠재적인 이웃들을 연결하여 발생한 사이클. (c) x 를 포함하는 새로운 MST를 구하는 한 스텝. (d) x 를 포함하는 최종 MST 트리

Fig. 2. (a) MST and Steiner point x . (b) A cycle generated by connection x with its potential neighbour points. (c) One step of finding New MST including x . (d) New MST including x .

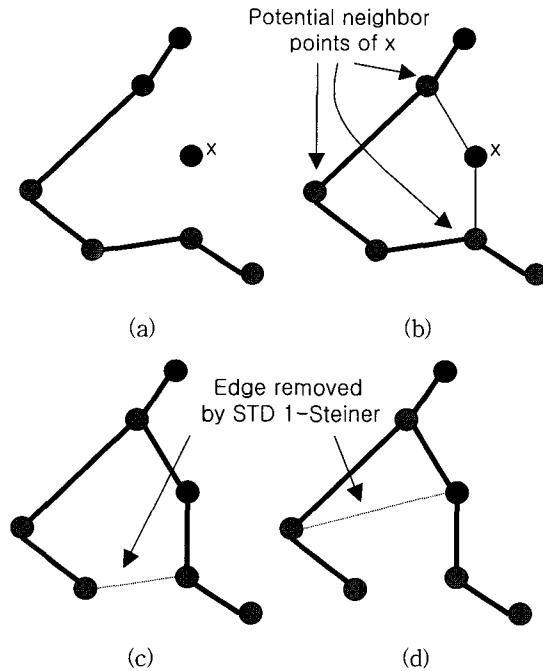


그림 3. (a) MST와 스타이너 점 x . (b) x 와 x 의 잠재적인 이웃들을 연결하여 발생한 사이클. (c) STD 1-Steiner 알고리즘에 의해 제거된 간선. (d) STD 1-Steiner 알고리즘에 의해 새롭게 갱신된 트리.

Fig. 3. (a) MST and Steiner point x . (b) A cycle generated by connection x with its potential neighbour points. (c) The edge removed by STD 1-Steiner algorithm. (d) New tree updated by STD 1-Steiner algorithm.

STD(Single-Source Timing Driven 1-Steiner Algorithm)
Input : Single-Source Net $P = \{ p_0, p_1, \dots, p_n \}$ with Source p_0
Output : Minimized Maximum Weighted Delay Tree of Net P
$T(P) \leftarrow MST(P)$ while SM 1-Steiner point z of P exists do $T(P) \leftarrow T(P \cup \{z\})$ /*Application of linear MST */ $P \leftarrow P \cup \{z\}$ end while return $T(P)$

그림 4. STD(Single-source Timing driven) 1-Steiner 알고리즘

Fig. 4. STD(Single-source Timing driven) 1-Steiner algorithm.

IV. 다중 소스 네트워크의 배선 알고리즘

1. 문제 정의

단일 소스 네트워크들이 단 하나의 입력 게이트(gate)와 하나 이상의 출력 게이트들로 구성되는 것에 비하여 다중 소스 네트워크는 하나 이상의 입력 게이트와 하나 이상의 출력 게이트들로 구성된다. 또 단일 소스 네트워크에서는 소스와 터미널의 역할을 하는 게이트들이 구분되어 있지만 다중 소스 네트워크에서 게이트는 소스이자 터미널이기도 하다. 따라서 단일 소스 배선 알고리즘을 그대로 다중 소스 네트워크를 배선하는데 적용하면 최적의 결과를 얻지 못할 가능성이 높다. 따라서 다중 소스 네트워크에 대한 별도의 배선 알고리즘을 고려할 필요가 있다.

다중 소스 네트워크 $P = \{ p_1, p_2, \dots, p_n \}$ 와 P 를 배선한 결과로 얻어진 트리 $T(P)$ 가 주어졌을 때 단일 소스와 같은 개념의 가중치 W_{ij} ($i \neq j$)를 각 소스 터미널 쌍 (p_i, p_j) 에 연계시킬 수 있다. W_{ij} 는 소스 p_i 와 터미널 p_j 사이의 지연시간 임계도(criticality)를 나타낸다. 또 소스 p_i 와 터미널 p_j 사이의 지연시간을 d_{ij} 라고 한다면 단일 소스 네트워크에서와 같이 동일하게 $W_{ij}d_{ij}$ 는 소스 p_i 와 터미널 p_j 사이의 가중 지연 시간을 나타낸다. 다중 소스 네트워크의 배선 목적은 네트워크의 최대 가중 지연 시간을 줄이면서 동시에 배선 길이를 줄이는 것이다. 정리하면 다음과 같다.

- (1) $\max \{ W_{ij}d_{ij} \}$ 를 최소화하기 ($p_i, p_j \in P, i \neq j$).
- (2) 전체 배선에 사용되는 와이어의 길이를 최소화하기.

본 논문에서 제안하는 MTD(Multi-source Timing Driven) 1-Steiner 알고리즘은 (2)의 목적보다 (1)의 목적에 더 우선 순위를 두고 배선을 수행한다. 가중치는 $[0, \infty]$ 사이에서 결정될 수 있으나 단일 소스 네트워크와 같이 프로그램의 간결성을 위해 각 소스, 터미널 쌍에 가중치를 1 또는 0을 할당한다. 즉 지연시간 임계도가 높은 소스, 터미널 쌍의 가중치는 1이고 그렇지 않은 소스, 터미널 쌍의 가중치는 0으로 한다.

2. MTD(Multi-source Timing Driven) 1-Steiner 알

고리즘

MTD 1-Steiner 알고리즘은 본 논문에서 제안하는 성능 지향형 다중 소스 네트워크 배선 알고리즘이다. MTD 1-Steiner 알고리즘은 1-Steiner 휴리스틱 알고리즘을 응용하여 반복적인 스타이너 점의 삽입을 통해 트리의 최대 가중 시간을 줄여간다.

MTD 1-Steiner 알고리즘을 기술하기 전에 MM(Multi-source) 1-Steiner 점을 정의하여 보다 효율적으로 알고리즘을 기술하도록 하겠다.

- MM 1-Steiner 점 : 다중 소스 네트워크 $P = \{ p_1, p_2, \dots, p_n \}$ 가 주어졌을 때 여기서 $MST(P)$ 를 P 의 MST라 하고, $mwd(T)$ 를 트리 T 의 최대 가중 지연 시간이라고 할 때, Hanan 그래프의 정점 z 중에서 아래의 조건 (1),(2),(3)을 동시에 만족하는 스타이너 점 z .

- (1) $z \notin P$, (2) $mwd(MST(P)) > mwd(MST(P \cup \{z\}))$, (3) $mwd(MST(P \cup \{z\}))$ 가 최소

다중 소스 네트워크 $P = \{ p_1, p_2, \dots, p_n \}$ 와 스타이너 점이 하나 주어졌을 때 MTD 1-Steiner 알고리즘은 먼저 P 와 주어진 스타이너 점의 MST를 구한다. 이 과정까지는 1-Steiner 알고리즘과 동일하다. 그러나 1-Steiner 알고리즘이 MST의 배선길이를 측정하는 것과 달리 MTD 1-Steiner 알고리즘은 MST의 최대 가중 지연 시간을 측정한다는 점에서 차이가 있다.

MTD 1-Steiner 알고리즘은 다음과 같이 진행된다. 다중 소스 네트워크 $P = \{ p_1, p_2, \dots, p_n \}$ 가 주어졌을

MTD(Multi-source Timing Driven) 1-Steiner Algorithm
Input : Multi-source Net $P = \{ p_1, p_2, \dots, p_n \}$
Output : Minimized Maximum Weighted Delay Tree of P
$T(P) \leftarrow MST(P)$
while MM 1-Steiner point z of P exists do
$P \leftarrow P \cup \{z\}$ /* update P */
$T(P) \leftarrow MST(P)$
end while
return $MST(P)$

그림 5. MTD(Multi-source Timing Driven) 1-Steiner 알고리즘

Fig. 5. MTD(Multi-source Timing Driven) 1-Steiner algorithm.

때 P 의 MM 1-Steiner 점 z 를 구한다. 그리고 $P \leftarrow P \cup \{z\}$ 로 갱신한다. 여기까지가 MTD 1-Steiner 알고리즘의 한 단계이다. 계속하여 더 이상 P 의 MM 1-Steiner 점이 존재하지 않을 때까지 위의 단계를 되풀이한다. 따라서 MTD 1-Steiner 알고리즘의 단계가 거듭 될수록 점점 $MST(P)$ 의 최대 가중 지연 시간이 작아지게 된다. 그림 5에 MTD 1-Steiner 알고리즘을 보인다.

V. 실험결과

본 논문에서 제안하는 STD 1-Steiner 알고리즘과 MTD 1-Steiner 알고리즘은 C 언어를 사용하여 구현하였으며, 512MB의 기억용량을 가진 펜티엄-III PC에서 실험하였다. 단위 길이 와이어의 저항과 정전용량, 게이트의 구동 저항(driver resistance) 및 다이오드의 부하 정전용량(loadng capacitance)은 $0.5\mu\text{m}$ CMOS IC 테크놀러지와 $0.18\mu\text{m}$ CMOS IC 테크놀러지 및 MCM 테크놀러지의 파라미터(parameter)를 이용하였다. $0.5\mu\text{m}$ CMOS IC 테크놀러지와 MCM 테크놀러지의 경우 [3,5,10]에서 사용한 값과 동일하며 $0.18\mu\text{m}$ CMOS IC 테크놀러지는 $0.5\mu\text{m}$ CMOS IC 테크놀러지를 스케일 하여 얻어졌다. 표 1에 실험에 사용된 테크놀러지가 나와 있다.

실험에 사용된 네트는 크기 별로 200개씩, 단위 그리드 간격이 $10\mu\text{m}$ 인 그리드 위에서 무작위로 발생시켜

얻었다. 이에 따라 표 1에 나와있는 칩의 크기(chip size)에 의해 IC 테크놀러지 파라미터를 적용하는 네트에 대해서는 1000×1000 그리드를, 그리고 MCM 테크놀러지 파라미터를 적용하는 네트에 대해서는 10000×10000 그리드를 사용하였다.

표 1. 실험에 사용된 테크놀러지 파라미터.
Table 1. Technology parameter used in Experiment.

Technology	$0.18\mu\text{m}$ CMOS IC	$0.5\mu\text{m}$ CMOS IC	MCM
unit wire resistance ($\Omega/\mu\text{m}$)	0.311	0.0463	0.002
unit wire capacitance (fF/ μm)	0.014	0.189	0.085
driver resistance (Ω)	750	270	25
loading capacitance (fF)	0.144	2.68	1000
chip size	$1\text{cm} \times 1\text{cm}$	$1\text{cm} \times 1\text{cm}$	$10\text{cm} \times 10\text{cm}$

1. STD 1-Steiner 알고리즘 실험결과

STD 1-Steiner 알고리즘에 관하여는 테크놀러지들에 따라 네트의 모든 터미널이 임계 터미널인 경우와 임계 터미널이 단 하나 존재하는 경우를 구분하여 실험하였다. 그리고 객관적인 STD 1-Steiner 알고리즘의 성능을 측정하기 위하여 1-Steiner^[1], A-Tree^[3], SERT^[4], SERT-C^[4], BB-SORT^[5], BB-SORT-C^[5] 알고리즘들을 이용하여 동일한 네트를 배선한 후 그 결

표 2. 모든 터미널이 임계 터미널인 단일 소스 네트들을 $0.18\mu\text{m}$ CMOS IC 테크놀러지를 적용하여 각 알고리즘 별로 배선한 결과 비교.

Table 2. Comparison of single-source net routing results through $0.18\mu\text{m}$ CMOS IC technology. The net is assumed to have all critical pairs.

Net size		4		8		12		16	
		value	ratio	value	ratio	value	ratio	value	ratio
Max Delay (ns)	1-Steiner	0.403	100	0.701	100	0.989	100	1.158	100
	A-Tree	0.378	93.8	0.588	83.9	0.734	74.3	0.859	74.1
	SERT	0.367	91.3	0.552	78.8	0.680	68.8	0.786	67.9
	BB-SORT	0.366	90.9	0.537	76.6	-	-	-	-
	STD 1-Steiner	0.367	91.1	0.545	77.8	0.659	66.7	0.762	65.8
Tree Length (cm)	1-Steiner	1.277	100	2.058	100	2.609	100	3.000	100
	A-Tree	1.277	100.0	2.131	103.6	2.782	106.6	3.251	108.4
	SERT	1.434	112.3	2.441	118.6	3.291	126.2	3.781	126.0
	BB-SORT	1.419	111.1	2.367	115.0	-	-	-	-
	STD1-Steiner	1.431	112.1	2.392	116.3	3.128	119.6	3.577	119.2

표 3. 임계 터미널이 단 하나 존재하는 단일 소스 네트들을 0.18 μ m CMOS IC 테크놀러지를 적용하여 각 알고리즘 별로 배선한 결과 비교

Table 3. Comparison of single-source net routing results through 0.18 μ m CMOS IC technology. The net is assumed to have only one critical pair.

Net size		4		8		12		16	
		value	ratio	value	ratio	value	ratio	value	ratio
Critical Delay (ns)	1-Steiner	0.332	100	0.611	100	0.752	100	0.884	100
	A-Tree	0.311	93.7	0.507	83.1	0.592	78.7	0.665	75.2
	SERT-C	0.285	85.8	0.457	74.8	0.558	74.1	0.665	75.2
	SORT-C	0.278	83.8	0.407	66.6	-	-	-	-
	STD 1-Steiner	0.281	84.6	0.415	67.9	0.456	60.6	0.505	57.1
Tree Length (cm)	1-Steiner	1.261	100	2.089	100	2.565	100	2.975	100
	A-Tree	1.261	100.0	2.162	103.5	2.740	106.8	3.215	107.8
	SERT-C	1.433	113.6	2.324	111.3	2.785	108.6	3.221	108.1
	SORT-C	1.474	116.5	2.433	116.5	-	-	-	-
	STD 1-Steiner	1.491	118.2	2.470	118.3	2.990	116.6	3.430	115.1

과를 비교하였다. 네트는 크기 별로 200개씩 무작위로 발생시킨 후 배선하였으며 각각 배선된 네트의 최대 가중 지연 시간과 배선의 길이, 그리고 실행시간을 측정하였다.

먼저 표 2는 모든 터미널이 임계 터미널인 네트를 크기 별로 200개씩 무작위로 발생시킨 후 0.18 μ m CMOS IC 테크놀러지를 이용하여 각 알고리즘 별로 배선한 결과를 보여준다. 표 2에서 각 측정값 바로 오른쪽 행의 비율(ratio)은 1-Steiner 알고리즘의 측정값에 대한 해당 측정값의 비율이다. 모든 터미널의 가중치가 동일하므로 최대 가중 지연시간은 최대 지연시간(max delay)과 같다. BB-SORT는 네트의 크기가

커지면 알고리즘을 실행하는데 걸리는 시간이 매우 길어진다. 표 2에서 12 이상의 크기를 가진 네트에 대하여 BB-SORT의 측정값을 생략한 이유가 바로 이것이다. 또 표 3은 임계 터미널이 단 하나 존재하는 네트를 크기 별로 200개씩 무작위로 발생시킨 후 0.18 μ m CMOS IC 테크놀러지를 이용하여 각 알고리즘 별로 배선한 결과를 보여준다. 임계 터미널은 터미널 중에서 무작위로 하나를 선택하였다. 표 3의 임계 지연시간(critical delay)이란 임계 터미널까지 지연시간을 말한다. 또 이번 실험은 임계 터미널이 하나 존재하는 네트를 배선하는 실험이므로 SERT 대신 SERT-C를 BB-SORT 대신 BB-SORT-C를 사용하여 결과를 비교

표 4. 모든 터미널이 임계 터미널인 단일 소스 네트들을 MCM 테크놀러지를 적용하여 각 알고리즘 별로 배선한 결과 비교

Table 4. Comparison of single-source net routing results through MCM technology. The net is assumed to have all critical pairs.

Net size		4		8		12		16	
		value	ratio	value	ratio	value	ratio	value	ratio
Max Delay (ns)	1-Steiner	1.679	100	3.491	100	5.324	100	6.344	100
	A-Tree	1.566	93.3	2.887	82.7	3.589	67.4	4.311	68.0
	SERT	1.439	85.7	2.460	70.5	3.036	57.0	3.574	56.3
	BB-SORT	1.416	84.3	2.284	65.4	-	-	-	-
	STD 1-Steiner	1.435	85.5	2.413	69.1	2.939	55.2	3.413	53.8
Tree Length (cm)	1-Steiner	13.047	100	20.550	100	25.901	100	30.642	100
	A-Tree	13.047	100.0	21.256	103.4	27.716	107.0	33.008	107.7
	SERT	17.232	132.1	36.952	179.8	54.261	209.5	71.793	234.3
	BB-SORT	17.442	133.7	30.813	149.9	-	-	-	-
	STD 1-steiner	16.615	127.4	27.268	132.7	33.593	129.7	39.128	127.7

표 5. 임계 터미널이 단 하나 존재하는 단일 소스 네트들을 MCM 테크놀러지를 적용하여 각 알고리즘 별로 배선한 결과 비교

Table 5. Comparison of single-source net routing results through MCM technology. The net is assumed to have only one critical pair.

Net size		4		8		12		16	
		value	ratio	value	ratio	value	ratio	value	ratio
Critical Delay (ns)	1-Steiner	1.606	100	2.730	100	3.352	100	4.597	100
	A-Tree	1.439	89.6	2.232	81.8	2.493	74.4	3.123	67.9
	SERT-C	1.135	70.6	1.652	60.5	1.776	53.0	2.149	46.7
	SORT-C	1.118	69.6	1.397	51.2	-	-	-	-
	STD 1-Steiner	1.134	70.6	1.461	53.5	1.575	47.0	1.937	42.1
Tree Length (cm)	1-Steiner	13.691	100	20.692	100	25.813	100	30.771	100
	A-Tree	13.691	100.00	21.600	104.4	27.501	106.5	33.137	107.7
	SERT-C	17.354	126.8	25.045	121.0	30.509	118.2	36.192	117.6
	SORT-C	17.763	129.7	25.984	125.6	-	-	-	-
	STD 1-Steiner	17.630	128.8	25.777	124.6	30.796	119.3	36.496	118.6

하였다.

다음 표 4와 표 5는 MCM 테크놀러지 파라미터를 가진 네트들을 배선한 결과이다. 실험 측정값 및 실험에 사용된 알고리즘은 표3~표4와 동일하다. 단 칩의 크기가 10cm×10cm라고 가정하고 10000×10000의 크기를 가진 그리드 위에서 네트를 크기별로 200개씩 무작위로 발생시켜 실험하였다.

표2~표5까지 실험에서 STD 1-Steiner 알고리즘을 제외한 다른 알고리즘 중 가장 좋은 성능을 보인 것은 SERT 알고리즘이다. SERT 알고리즘에 비하여 STD 1-Steiner 알고리즘은 네트의 모든 터미널이 임계터미널 일 때 평균 2.1%, 최대 4.5% , 임계 터미널이 단 하나

존재할 때는 평균 10.6%, 최대 24.1 % 지연 시간이 감소하는 성능을 각각 보였다. 따라서 STD 1-Steiner 알고리즘이 기존의 배선 알고리즘 보다 최근의 IC CMOS, MCM 테크놀러지를 사용하는 회로에 더 효과적으로 적용할 수 있음을 알 수 있다.

2. MTD 1-Steiner 알고리즘 실험결과

MTD 1-Steiner 알고리즘도 STD 1-Steiner 알고리즘과 동일하게 다중 소스 네트를 무작위로 200개씩 발생한 후 표 1의 테크놀러지 파라미터를 적용하여 배선을 수행하였다. 그리고 네트의 모든 터미널이 임계 터미널인 경우와 모든 터미널 중에서 임계 터미널이 일부분 존재하는 경우 2가지를 가정하여 실험하였으며,

표 6. 0.5μm CMOS IC 테크놀러지 파라미터를 이용한 MTD 1-Steiner 알고리즘과 MCMD A-Tree 알고리즘의 비교

Table 6. Comparison of MTD 1-Steiner algorithm with MCMD A-Tree algorithm through 0.5μm CMOS IC technology parameter.

Net Size	Algorithm	Max Delay	Average Max Delay	Average All Delay	Tree Length
4	MCMD A-Tree	98.9	98.8	99.1	100
	MTD 1-Steiner	98.6	99.5	99.9	100
8	MCMD A-Tree	94.1	98.1	97.5	100.2
	MTD 1-Steiner	94.6	97.9	99.6	101.5
16	MCMD A-Tree	86.3	93.6	96.6	104.6
	MTD 1-Steiner	89.6	95.4	99.8	103.6

unit : %

표 7. MCM 테크놀러지 파라미터를 이용한 MTD 1-Steiner 알고리즘과 MCMD A-Tree 알고리즘의 비교

Table 7. Comparison of MTD 1-Steiner algorithm with MCMD A-Tree algorithm through MCM technology parameter.

Net Size	Algorithm	Max Delay	Average Max Delay	Average All Delay	Tree Length
4	MCMD A-Tree	96.5	98.3	100	100
	MTD 1-Steiner	98.7	99.6	100	100
8	MCMD A-Tree	87.2	92.6	100.3	99.6
	MTD 1-Steiner	94.1	97.7	100.3	101.7
16	MCMD A-Tree	78.9	88.5	96.8	104.4
	MTD 1-Steiner	83.7	91.7	98.2	104.6

unit : %

MTD 1-Steiner 알고리즘의 객관적인 성능을 평가하기 위하여 MCMD A-tree^[10]의 결과¹⁾를 그대로 인용하여 MTD 1-Steiner 알고리즘과 성능 비교를 했다.

표 6은 네트의 크기에 따른 MCMD A-Tree 알고리즘과 MTD 1-Steiner 알고리즘의 결과를 보여준다. 표 6에서 최대 지연시간(Max Delay)은 모든 소스, 터미널 쌍간의 지연시간 중 최대값을 가지는 지연시간이다. 그리고 최대 지연시간의 평균(Average Max Delay)은 각 소스에서 각 터미널까지의 지연시간 중 최대의 지연시간의 평균이다. 그리고 모든 지연시간의 평균(All Delay Average)은 모든 소스, 터미널사이의 지연시간의 평균이다. 그리고 트리의 길이(Tree Length)는 배선에 사용된 와이어의 길이이다. 표 7은 표 6과 동일한 실험환경에서 실험한 결과이다. 단 표 6의 네트는 0.5 μm CMOS IC 테크놀러지 파라미터를 적용하여 배선하였고, 표 7의 네트는 MCM 테크놀러지 파라미터를 적용하여 배선하였다. 또 표 8은 크기가 8인 네트에서 임계 소스, 터미널 쌍이 1, 2, 3, 10 존재하는 경우를 가정하고 0.5 μm CMOS IC 테크놀러지 파라미터를 적용하여 배선한 결과이다.

1) Cong 등^[10]은 spice 모델을 이용하여 실험 결과를 제시하였다. 본 연구에서는 별도의 실험을 통해 1-Steiner 알고리즘의 Elmore 지연시간에 대한 MTD 1-Steiner 알고리즘의 Elmore 지연시간의 비율과 1-Steiner 알고리즘의 spice 모델 지연시간에 대한 MTD 1-Steiner 알고리즘의 spice 모델 지연시간 비율이 거의 동일한 결과를 얻었다.

표 6~표 8에서 MTD 1-Steiner는 1-Steiner 알고리즘의 Elmore 지연시간에 대한 MTD 1-Steiner 알고리즘의 Elmore 지연시간의 비율을 나타내고 MCMD A-tree는 1-Steiner 알고리즘의 spice 지연시간에 대한 MCMD A-tree 알고리즘의 지연시간의 비율을 나타낸다.

표 6과 표 7의 결과에서 MTD 1-Steiner 알고리즘은 MCMD A-tree 알고리즘의 결과와 비교하여 최대 지연시간이 최대 6.8% 큰 결과를 보여 준다. 그에 반하여 표 8은 MTD 1-Steiner 알고리즘이 MCMD A tree 알고리즘보다 임계 소스, 터미널 사이의 최대 지연시간 면에서 최대 5.3% 작은 것을 보여준다. 이렇게 네트에 임계 소스, 터미널이 전체적으로 존재할 때와 일부분 존재할 때 성능의 차이가 있는 데에는 두 가지 이유가 있다. 첫 번째는 MTD가 Hanan 그래프를 기본으로 배선을 하기 때문이다. 다중 소스 네트의 경우 네트의 핀의 분포에 따라서 Hanan 그래프의 정점만으로는 배선을 하는 데 불충분한 경우가 존재한다. 참고로 MCMD A-tree는 Non-Hanan 환경에서 배선을 수행한다. 두 번째는 MTD 1-Steiner 알고리즘이 MST를 사용하여 배선을 수행하기 때문이다. MST를 사용하여 배선을 할 경우 일부 핀 사이의 거리를 단축하여 지연시간을 줄이는데는 비교적 효과적이거나 전체 핀 사이의 거리를 단축하는데는 MST가 제약을 많이 주기 때문이다. 따라서 네트에 임계 소스, 터미널 쌍이 부분적으로 존재할 때 MTD 1-Steiner 알고리즘은 배선을 효과적으로 수행하지만 네트에 전체 소스, 터미널 쌍이 임계 소스, 터미널일 경우 최대 지연시간을 최소화하는데는 비교

표 8. 네트에 부분적인 임계쌍이 존재할 때 MTD 1-Steiner 알고리즘과 MCMD A-Tree 알고리즘의 비교

Table 8. Comparison of MTD 1-Steiner algorithm with MCMD A-Tree algorithm when net has a subset of critical pairs.

Critical pair num	Algorithm	Max Delay	Average Max Delay	Average All Delay	Tree Length
1	MCMD A-Tree	85.8	85.8	85.8	97.3
	MTD 1-Steiner	88.4	88.4	88.4	103.1
2	MCMD A-Tree	95.1	99.3	98.6	99.6
	MTD 1-Steiner	90.0	92.1	95.4	103.7
3	MCMD A-Tree	96.3	100	100	99.7
	MTD 1-Steiner	93.3	97.2	97.2	103
10	MCMD A-Tree	94.0	101.5	100.7	99.5
	MTD 1-Steiner	93.7	99.2	99.4	102.1

unit : %

적 그 성능이 떨어진다.

VI. 결 론

본 논문에서는 1-Steiner 휴리스틱 알고리즘을 응용하는 단일 소스 배선 알고리즘 STD 1-Steiner 와 다중 소스 배선 알고리즘 MTD 1-Steiner를 제안하였다. 본 논문에서 제안하는 알고리즘들은 반복적인 스타이너 점 삽입과 트리의 갱신을 통하여 전 단계의 트리보다 최대 가중 지연 시간을 줄여나가는 성능 지향형 알고리즘이다.

실험에서 STD 1-Steiner 알고리즘은 기존의 단일 소스 배선 알고리즘의 결과와 비교하였을 때 SERT[4]에 비하여 2.1% SERT-C[4]에 비하여 10.6% 정도의 지연 시간 감소를 보였다. 또 MTD 1-Steiner 알고리즘은 MCMD A-Tree와 비교했을 때 모든 소스, 터미널 쌍이 임계쌍(critical pair)일 경우는 최대 지연 시간이 평균 2.7% 증가했지만 부분적인 임계쌍이 존재할 때는 최대 지연 시간이 평균 1.4% 감소하는 성능을 보였다. 이에 따라 1-Steiner 휴리스틱을 응용하는 본 논문의 배선 방법들이 단일 소스 네트와 다중 소스 네트에서 효과적으로 성능 지향형 배선을 수행하는 것을 확인하였다.

앞으로의 계획은 본 논문의 배선 방법을 Hanan 환경의 제약을 넘어 Non-Hanan 점을 포함하는 배선 방법

으로 발전시키는 것과, 또 버퍼 삽입 및 와이어의 너비 조절을 도모하는 배선 방법으로 발전시키는 것이다. 그리고 MTD 1-Steiner 알고리즘에서 사용하는 MST (Minimum Spanning Tree)보다 더 효과적으로 최대 가중 지연 시간을 줄여 주는 알고리즘을 개발하는 것이다.

참 고 문 헌

- [1] A. B. Kahng, G. Robins, "A New Class of Iterative Steiner Tree Heuristics with Good Performance," IEEE Trans. Computer-Aided Design, Vol. 11, pp. 893~902, July 1992.
- [2] D. Richards, "Fast Heuristic Algorithms for Rectilinear Steiner Trees," Algorithmica 4, pp. 191~207, 1989.
- [3] J. Cong, C. K. Koh, "Performance-Driven Interconnect Design Based on Distributed RC Delay Model," in Proc. ACM/IEEE Design Automat. Conf., 1993, pp. 606~611.
- [4] K. D. Boese, A. B. Kahng, G. Robins, "High Performance Routing Trees with Identified Critical Sinks," in Proc. ACM/IEEE Design Automat. Conf., 1993, pp. 182~187.
- [5] K. D. Boese, A. B. Kahng, G. Robins, "Near-

- optimal Critical Sink Routing Tree Constructions," *IEEE Trans. Computer-Aided Design*, Vol. 14, pp. 1417~1436, Dec. 1995.
- [6] H. Hou, J. Hu, S. S. Sapatnekar, "Non-Hanan Routing," *IEEE Trans. Computer-Aided Design*, V.18, pp. 436~444, April 1999.
- [7] W. C. Elmore, "The Transient Response of Damped Linear Network with Particular Regard to Wideband Amplifier," *J. Applied Physics*, pp. 55~63, 1948.
- [8] K. D. Boese, A. B. Kahng, B. A. McCoy, G. Robins, "Fidelity and Near-Optimality of Elmore-Based Routing Constructions," in *Proc. IEEE Intl. Conf. on Computer Design*, 1993, pp. 81~84.
- [9] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, "A Direct Combination of The Prim and Dijkstra Constructions for Improved Performance-Driven Routing," in *Proc. IEEE Int. Symp. on Circuits Syst.*, 1993, pp. 1869~1872.
- [10] J. Cong, P. H. Madden, "Performance Driven Routing with Multiple Sources," *IEEE Trans. Computer-Aided Design*, V.16, pp. 410~419, April 1997.
- [11] J. Lillis, C. K. Cheng, "Timing Optimization for Multisource Nets: Characterization and Optimal Repeater Insertion," *IEEE Trans. Computer-Aided Design*, V.18, pp. 322~331, March 1999.
- [12] J. Cong, L. He, "Optimal Wiresizing for Interconnects with Multiple Sources," *ACM Trans. Design Automation of Electronic Systems*, Vol. 1, pp. 478~511, Oct. 1996.
- [13] A. B. Kahng, G. Robins, *On Optimal Interconnections for VLSI*. Norwell, MA : Kluwer Academic, 1995.
- [14] M. Garey, D. S. Johnson, "The Rectilinear Steiner Problem is NP-Complete," *SIAM J. App. Math.*, 32(4), pp. 826~834, 1977.
- [15] M. Hanan, "On Steiner's Problem with Rectilinear Distance," *SIAM J. App. Math.* 14, pp. 255~265, 1966.

 저 자 소 개

沈 昊(正會員)

1999年 서강대학교 물리학과 학사. 2001年 : 서강대학교 컴퓨터학과 석사. 2001年~현재 : 삼성전자 반도체 연구소 CAE팀 연구원

林 鍾 錫(正會員)

1981年 : 서강대학교 전자공학과 학사. 1983年 : 한국과학기술원 전기 및 전자공학과 석사. 1989年 : Univ. of Maryland, College Park, 전기공학과 박사. 1983年 3月~1990年 8月 : 한국전자통신연구소 연구원. 1990年 9月~현재 : 서강대학교 컴퓨터학과 교수