論文95-32B-8-2

# 연합 처리기를 이용한 직교선형 스타이너 트리의 병렬 알고리즘

# (A Parallel Algorithm For Rectilinear Steiner Tree Using Associative Processor)

朴 台 根 *

(Taegeun Park)

요 약

본 논문은 연합 처리기를 이용하여 Minimum Spanning Tree(MST)로부터 직교선형 스타이너 트리를 구성하는 방법에 관하여 기술하였다. 본 연구에서 Content-Addressable Memory(CAM)의 선택적인 매칭 기능과 단순한 처리능력을 가진 논리회로로써 구현된 연합처리기의 기본 알고리즘을 이용한 병렬 알고리즘을 제안하였다. 제안된 알고리즘의 중심개념은 MST 내의 인접한 에지(edge)들 간에 중복부분을 최대화시킴으로써 구성되는 직교선형 스타이너 트리의 길이를 최소화하는 것이다. n이 노드(node)의 갯수일 때, 효과적인 MST 구성 알고리즘을 이용하여 결과적으로 선형적인 O(n)의 직교선형 스타이너 트리 구성 알고리즘을 제안하였다. Z-형태 배열을 가능하게 하기 위하여 노드 삽입 방법이 제안되었다. 결과적으로, 인접한 에지에만 국한되고 재구성이 불필요한 구현방법을 통하여 직교선형 스타이너 트리 구성속도가 향상되었다.

**Abstract**

This paper describes an approach for constucting a Rectilinear Steiner Tree (RST) derivable from a Minimum Spanning Tree (MST), using Associative Processor (AP). We propose a fast parallel algorithm using AP's basic algorithms which can be realized by the processing capability of rudimentary logic and the selective matching capability of Content-Addressable Memory (CAM). The main idea behind the proposed algorithm is to maximize the overlaps between the consecutive edges in MST, thus minimizing the cost of a RST. An efficient parallel linear algorithm with O(n) complexity to construct a RST is proposed using an algorithm to find a MST, where n is the number of nodes. A node insertion method is introduced to allow the Z-type layout. The routing process which only depends on the neighbor edges and the no-rerouting strategy both help to speed up finding a RST.

## I. 서 론

각 에지(edge)가 양의 가중치를 갖는 일반적인 비방향성 연결그래프(undirected connected graph) G를 구성하는 노드(node)들의 유한집합 P를 생각해 보자. 평면상의 점들의집합 P에 대한 직교선형 스타이너 트리 T는 집합 P를 연결하는 수평과 수직의 선들을 구성하는 노드들의 집합 N을 갖는 트리라고 정의할 수 있다. 집합 P의 점들은 종결점(terminal points) 그리고 N-P의 점들은 스타이너 점(Steiner points)이라고 부른다. 스타이너 트리의 가중치는 T를 구성하는

* 正會員, 現代電子産業 (株) 半導體 第2研究所
(Hyundai Elec. Ind. Co. Ltd., Semiconductor Lab. 2)

接受日字:1994年6月21日, 수정완료일:1995年8月4日

모든 에지들의 가중치 합이라고 정의된다. 직교선형 스타이너 트리 구성문제는 NP-complete 문제이다[2] 지금까지 많은 연구가 수행되었고, 여러 가지 경험적 접근방법들이 제안되었다. Hwang[11]은 직선형 Minimum Spanning Tree(MST) 값과 최적 직교선형 스타이너 트리 값과의 비율은 3/2보다 크지 않다는 것을 증명하였다. 그러므로 많은 직교선형 스타이너 트리 구성을 위한 경험적 알고리즘들이 MST로부터 해답을 찾는다[6,7,8] Wong[7]은 직교선형 스타이너 트리 구성에 있어서 안정도(stability), 즉, 트리의 길이를 변화시키지 않으며 구성 선분을 재배치할 수 있는 성질을 찾아내었다. 또한, 임의의 계단형 배선을 가능하게 하는 다항시간 알고리즘을 구성하였다. Sakai[8]는 Delaunay네트 상에서 최소트리(minimal tree)의 간격 그래프(interval graph)를 이용하여 직교선형 스타이너 트리 구성을 접근하였다. 기존의 방법과 비교하여 본 연구는 MST를 구성하는 에지들에 대하여 병렬적으로 검색하고 상응하는 최적의 직교선형 스타이너 트리를 구성한다. 선형적인 알고리즘의 복잡도와 실제 하드웨어에서의 실험으로 상당히 빠른 구성속도를 보여주었다.

스타이너 트리 문제는 컴퓨터 네트워크와 VLSI 설계의 전반적인 배선시 경로를 결정하는 데 사용되어 왔다[5]. 즉, 총 회로연결 길이를 최소화하기 위하여 다항시간 직교선형 스타이너 트리 알고리즘(polynomial-time algorithm)을 이용하였다. 직교선형 스타이너 트리는 또한 다른 VLSI 설계 응용분야, 예를 들면 각 표준셀의 배치시에 회로연결 길이를 추정하는 데에도 이용된다[3,4] Gilbert[9]는 스타이너 트리를 통신망 설계에 응용하였다.

본 논문에서는 Content-Addressable Memory (CAM)를 이용하여 직교선형 스타이너 트리 구성을 위한 병렬알고리즘을 제안하였다. CAM은 현재 CMOS등의 VLSI기술로 구현될 수 있다. 각 CAM 워드에 간단한 처리논리회로를 첨가함으로써 CAM은 연합처리기가 된다. 검색 위주의 응용분야에서 CAM은 상당히 효율적인 해답이 될 수 있다. 또한 CAM 구조는 가중치를 갖는 에지와 노드의 집합으로 구성되는 그래프를 저장하기에 아주 적합하다. CAM에 저장된 노드의 정보를 검색함으로써 각 에지들의 연결에 대한 정보를 쉽게 찾을 수 있다. MST 구성시에는 CAM의 선택적 매치(selective match) 기능을 이용한 bit-serial word-parallel 방법으로 최소 가중치를 갖는 에지를 쉽게 찾을 수 있다. MST구성 후에는 MST의 종점(leaf node)으로부터 직교선형 스타이너 트리를 완성할 때까지 순차적으로 에지들을 해체하여 나간다. 총 배선길이를 감소시키는 Z-형태 배열을 허용하기 위하여 노드 삽입 방법이 제안되었다.

본 직교선형 스타이너 트리 구성 알고리즘의 특징은 다음과 같이 요약할 수 있다.

1) 선형적인 직교선형 스타이너 트리 구성 알고리즘으로 구성속도가 향상되었고 구성시 각 MST 에지들의 형태를 분류하여 처리함으로써 직교선형 스타이너 트리의 결과를 향상시켰다.

2) CAM과 간단한 처리능력을 가진 로직으로 이루어진 SIMD 구조의 연합처리기를 이용하여 직교선형 스타이너 트리 알고리즘을 병렬화하였다. 알고리즘은 bit-serial word-parallel로 동작한다. 트리 형태의 구조를 갖는 그래프를 CAM에 저장하여 처리하는 방법이 제시되었다.

3) 노드삽입을 이용하여 Z-형태의 배열을 허용함으로써 직교선형 스타이너 트리 결과가 향상되었고 인접 에지에만 국한되고 재구성이 불필요하므로 구성속도가 향상되었다.

2장에서는 본 연구에 사용된 연합처리기 하드웨어에 대하여 언급하였고, 3장에서는 직교선형 스타이너 트리 알고리즘이 기술되었고, 실험결과는 4장에서 논의되었다.

## Ⅱ. 연합처리기 하드웨어 구조

CAM 워드는 주소에 의해서가 아니라 그에 저장된 내용에 의해서 선택된다. 메모리의 각 워드에 간단한 처리논리회로를 첨가함으로써 CAM은 연합처리기가 될 수 있다. 메모리의 각 워드는 이전에 수행된 검색결과나 그것들을 이용한 논리수행을 기초로 하여 읽혀지고 쓰여질 수 있다. 연합처리기 형태의 하드웨어 구조는 von Neumann 병목현상을 경감시키는 프로세서와 메모리를 병행한 개념의 하드웨어 구조를 총칭한다고 할 수 있다.

본 연구는 IBM PS2/70 컴퓨터에 4096열(row)의 연합처리기[10]를 삽입한 하드웨어에서 수행되었다. 하드웨어는 memory-mapped device의 형태로 micro channel bus를 통해 주 컴퓨터와 접속되었다. 그림1

은 연합처리기의 구조도이다. 연합처리기는 그 기능상으로 제어부, 메모리, 로우 로직(row logic)으로 구분된다. 제어 부분은 세개의 36-bit 레지스터와 연합처리기에 의해 수행되는 마이크로코드를 저장하는 Writable Control Store로 구성되어 있다. 세개의 레지스터는 데이타 레지스터, 마스크 레지스터 그리고 네버매치(never-match) 레지스터이다.

연합처리기는 SIMD형태의 병렬컴퓨터 구조를 갖는다. 연합처리기 내에는 순차적 컴퓨터에서와 같이 하나의 명령어 처리기를 갖지만 그것과 연결된 수 많은 단순화된 1-비트 ALU가 각 CAM 워드마다 존재한다. 또한 각 CAM 워드는 주소의 개념이 없고 그 내용과 외부에서 주어진 값과 매치된 결과에 의하여 선택, 처리된다. 순차적 컴퓨터에서는 선택된 메모리 하나에 대하여 읽고, 쓰고, 처리하지만 연합처리기에서는 선택된 다수의 메모리가 동시에 수행되는 명령어에 의해서 영향을 받는다.

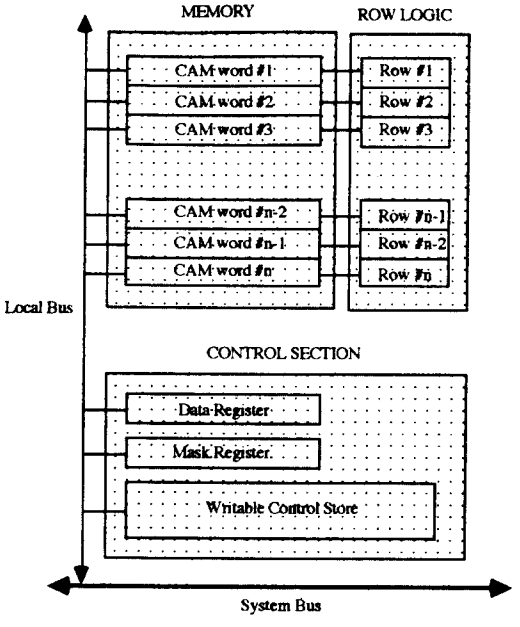


그림 1. 연합처리기의 하드웨어 구조
Fig. 1. Hardware structure of the AP.

그림2는 하나의 연합처리기소자의 구조를 나타낸다. 각 CAM 워드는 32 비트이고, 4 비트의 태그(tag) 영역을 가지고 있다. 각 태그 비트는 데이타 비트의 기능에 덧붙여 선택적으로 쓰여질 수도 있다. 태그 비트도 검색의 대상이므로 이를 적당히 변경함으로써 CAM에 저장하는 정보를 분류하여 처리 할 수 있다. 로우 로직은 General Purpose Logic Blok(GPLB), Multiple Response Resolver(MRR) 그리고 1-비트 레지스터인 R1, R2, R3로 구성된다. 이 논리회로들은 매치 결과를 저장하고 그것들을 처리하는 데 쓰인다. 만일 검색 중 하나 이상의 매치 결과가 발생했을 경우, MRR은 모든 매치 결과들을 우선순위에 따라 가장 낮은 주소의 것을 선택한다. GPLB는 일반적인 세 변수의 Boolean함수를 수행할 수 있다. 일반적인 RAM과 달리 CAM 워드들은 순차적인 주소의 개념을 갖지 않는 대신 각 워드는 선택 라인(select line)을 갖는다. 만일 선택라인이 활성화되었다면 그 워드는 수행되는 명령어에 의해 영향을 받는다. 열을 선택하기 위하여 GPLB 출력을 사용함으로써 동시에 여러 워드에 같은 정보를 쓸 수 있다. MRR는 반응벡터(response vector)를 저장하고, 필요하면 한번에 한 워드씩 명령을 수행할 수 있다(즉, 반응벡터 중 가장 상위에 위치하는 bit에 소속된 열만이 활성화된다). Shift Register(SR)는 연합처리기의 열 간의 통신에 사용되고 그 출력은 R1, R2, R3 중 하나로 입력될 수 있다. 연합처리기는 수천 개의 처리소자를 갖는 SIMD (single-instruction, multiple-data stream) 컴퓨터로써 이해될 수 있다.

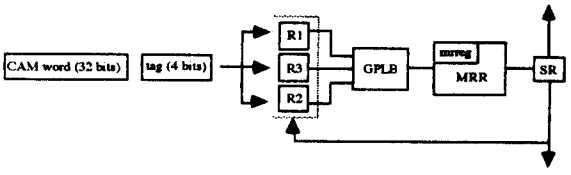


그림 2. 하나의 연합처리기소자 구조
Fig. 2. One Associative Processing element structure.

## Ⅲ. 직교선형 스타이너 트리 구성 알고리즘

가중치를 갖는 비방향성 그래프에 대한 MST는 모든 노드들을 연결하고 가중치의 총합이 최소화되는 에지들의 부분집합이라고 정의할 수 있다. MST 구성을 위한 Prim-Dijkstra 알고리즘은 n개의 노드가 있을 때 n2에 비례하는 수행시간을 필요로 한다. 본 연구는 CAM의 검색기능과 그 결과에 대한 간단한 논리 수행을 이용하여 복잡도가 O(n)인 MST를 구성하는 병렬 알고리즘을 이용하여 최종 직교선형 스타이너 트리 구

성에 있어서 수행속도와 알고리즘의 복잡도에서 커다란 향상을 보여주었다.

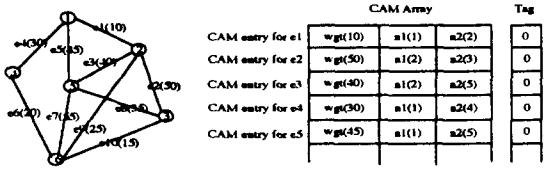


그림 3. 예제 그래프와 그의 CAM 정보
Fig. 3. Sample graph and its CAM entry.

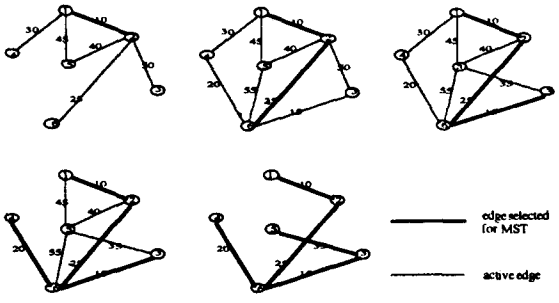


그림 4. 그림3의 예제에 대한 MST 구성 순서
Fig. 4. MST procedure for the sample graph in Fig. 3.

그림3은 간단한 그래프 표본과 그에 상응하는 CAM에 기재된 정보이다. 각 에지에 대한 정보는 한 쌍의 워드로 CAM에 저장된다. 첫번째 워드에는 에지를 구성하는 두개의 노드를 저장하고, 두번째 워드에는 그 에지에 대한 가중치를 저장한다(가중치는 부동 소수점 숫자일 수도 있다).

이제는 bit-serial word-parallel 방식으로 어떻게 병렬로 최소값을 찾을 수 있는지 설명하겠다. 먼저, 모든 워드들을 후보(candidate)로 최상위 bit(MSB)부터 각 비트에 대해 "0"을 검색한다. 각 단계마다 현재 비트가 "1"을 갖는 워드들은 후보에서 제외한다. 만일 현재 비트가 "0"인 워드가 하나도 없다면, 아무것도 제외하지 않고 다음단계로 내려간다. 제안된 연합처리기는 매치의 갯수가 하나일 때와 여러 개일 때를 구분할 수 있으므로 그의 기능을 이용할 수 있다. 결과적으로 마지막에 반응벡터에는 하나 또는 여러 개의 후보가 남아있게 된다. 이때 남아있는 후보들은 모두 같은 값을 가질 것이다. 부동소수점 숫자에 대해서는 지수부분을 먼저 검색하고 그 결과에 대해 가수부분을 검색하여 최소값을 찾는다.

그림4는 그림3의 예제에 대한 MST 구성의 순서이다. 각 반복(iteration)마다 하나의 새로운 노드가 찾아지고, 노드의 갯수(n) 만큼의 반복 후에는 MST가 구성된다[11].

제안된 알고리즘의 중심개념은 이미 구성된 MST 에지들 간에 중복부분을 최대화 시킴으로써 결과적으로 최적의 직교선형 스타이너 트리를 구성하는 것이다. 평면상에 n개의 노드가 있을 때의 직교선형 스타이너 트리문제는 주어진 노드들을 모두 연결하고 가중치의 합이 최소가 되는 직교선형 트리를 찾는 것이다. MST 에지 e=(i,j)의 두 종결점 i와 j 사이의 직교선형 최단경로는 그 에지의 계단형 배열(staircase layout)이라 불리운다. MST 내의 각 에지에 대한 계단형 배열의 경우를 찾아 에지들 간에 중복부분이 있을 경우 그 부분을 병합하게 될 때 그 결과의 구조는 바로 직교선형 스타이너 트리가 될 것이다. 최소 가중치의 트리를 얻기 위해서는 중복부분을 최대화 하는 계단형 배열을 선택해야 한다.

MST를 구성한 후 구성된 MST의 각 종점(leaf node)으로부터 시작하여 중복부분을 최대화 하는 방향으로 MST 에지를 해체하며 알고리즘이 진행되고, MST 에지들이 모두 해체되었을 때 그 결과는 직교선형 스타이너 트리가 된다. 제안된 알고리즘은 break-simple(BS)과 break-complex(BC)의 두 부분으로 구성된다. 그림5의 경우에서 P1이 시작점, P2가 중간점, P3가 다음점이라고 가정하자. BS부분은 하나의 에지가 중간점을 통하여 하나의 에지로 연결되는 경우를 처리한다. 예를 들어, P1이 시작점, 즉 종점이라고 가정하자. CAM의 에지정보를 검색함으로써 P2와 P3를 찾은 후, edge(P1,P2)와 edge(P2,P3) 사이의 중복부분을 최대화 하는 최적 배열을 찾는다.

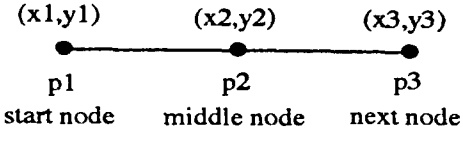


그림 5. 노드 표현 방식
Fig. 5. node representation.

그림6은 노드 삽입의 경우를 설명하고 있다. 새로운 스타이너점 P2'는 CAM에 저장되어 있는 edge

(P2,P3)는 edge(P2′,P3)로 대치된다. 새로운 edge(P2′,P3)는 이후 배열과정에서 이미 배열된 edge(P2,P2′)에 관계없이 독립적으로 배열될 수 있으므로 에지의 Z-형태 배열이 가능하다. 이러한 노드 삽입방법은 주어진 상황에서 배열을 최적화한다. 만일 중복부분을 찾을 수 없다면 그 에지에 대한 L-타입 배열이 가능하게 되고 노드 삽입은 불필요하게 된다. BS과정 중 삽입될 수 있는 노드는 각 에지당 많아야 하나이며 위치는 그림6에서 보는 바와 같이 y-좌표는 P1과 같으며 edge(P2,P3) 상에 있는 노드가 된다.

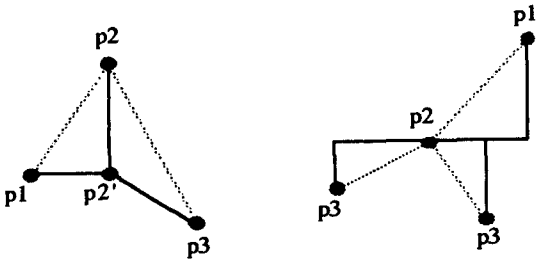


그림 6. 노드 삽입 방법
Fig. 6. node insertion method.

하나의 에지 배열이 끝나면 그 에지는 비활성화되어 이후의 검색과정에서 제외된다. 한번 에지가 배열되면 그 에지에 대한 재배열과정은 필요치 않다. 이제 P2나 P2′(노드 삽입의 유무에 따라)는 시작점이 되고, P3는 중간점 그리고 CAM 검색에 의하여 다음점을 찾는다. 만일 다음점이 하나라면, 위에 설명된 과정을 통해 edge(P2,P3)나 edge(P2′,P3)에 대한 배열을 찾는다. 만일 다음점이 둘 이상이면 시작점은 BC과정을 위해 스택(stack)에 저장된다. 모든 종점에 대하여 BS과정을 거쳐서 간단한 경우에 대한 배열을 완성한다.

BS과정이 끝나면, 과정중 스택에 저장된 노드에 대한 에지 배열을 찾기 위한 BC과정을 수행한다. 그림7에 예시된 경우에서와 같이 각 가능한 배열에 대한 x와 y축의 중복부분을 비교하여 최적의 배열을 찾는다. 배열이 끝나면 P3의 노드를 BS과정을 위해 스택에 저장한다. 모든 컴플렉스 노드들에 대한 배열이 끝나면 초기 MST를 구성하는 모든 에지가 완전히 배열될 때까지 BS와 BC과정을 반복한다. 구성과정중에 초기 MST는 각 종점으로부터 해체되어 최종적으로 직교선형 스타이너 트리를 구성하게 된다. 최적의 배열을 찾는 과정은 이웃 에지에만 관련되고, 재배열이 불필요하기 때문에 구성시간이 매우 빠르다. 또한 Z-형태 배열을 허용하므로 초기 MST와 직교선형 스타이너 트리간의 가중치 합의 비율이 매우 좋게 나타났다. 다음은 직교선형 스타이너 트리 알고리즘과 그의 복잡도를 분석하였다.

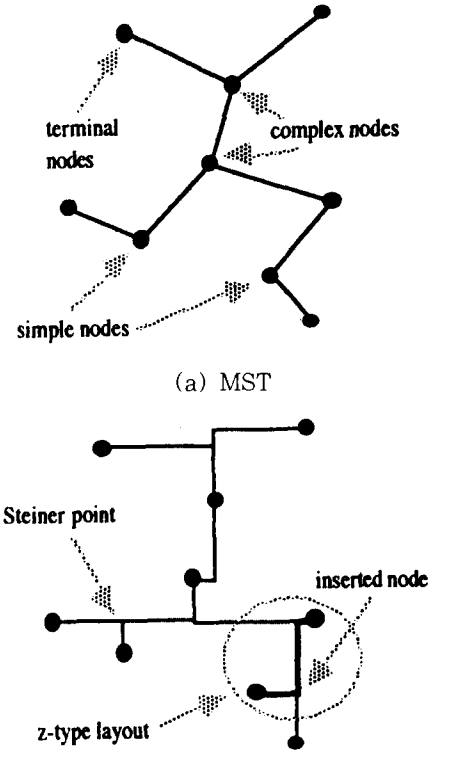


그림 7. 컴플렉스 타입의 배열
Fig. 7. Complex case example.

(Theorem) 제안된 직교선형 스타이너 트리 구성 알고리즘의 복잡도는 n이 노드의 갯수일 때 O(n)이다.

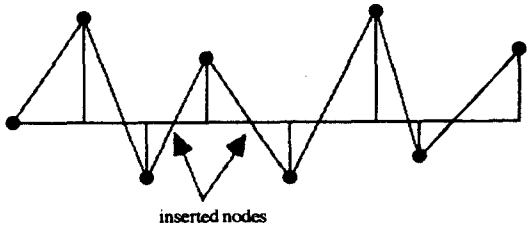


그림 8. 각 에지마다 노드를 삽입해야 하는 경우
Fig. 8. The worst node insertion case.

(Proof) 제안된 알고리즘은 O(n)의 알고리즘으로 구성된 n-1 개의 MST 에지들에 대하여 각 에지를 매 반복마다 해체하여 직교선형 스타이너 트리를 구성한다. 최악의 경우 각 에지마다 노드를 삽입할 경우 반복의 수는 2n-3 이 되며 재배선이 이루어지지 않으므로 결론적으로 최종 복잡도는 O(n)이다. 그림8은 각 에지마다 노드를 삽입해야 하는 경우이다.

알고리즘 직교선형 스타이너 트리
1. MST의 각 종점을 찾는다.
Repeat 모든 에지가 배열될 때 까지
2. Break-simple( ) 실행 반복
/ * 모든 simple 에지에 대하여 * /
3. Break-complex( ) 실행 반복
/ * 모든 complex 에지에 대하여 * /
End

Break-simple( )
1. 에지의 형태를 결정한다.
2. 중복부분이 최대화되도록 에지의 배열을 결정한다.
3. 필요하면 스타이너 점을 삽입하고 CAM의 에지를 변경한다.
4. 다음 노드를 찾는다.
5. 다음 노드가 simple 경우인지 complex 경우인지 결정한다.
만약 simple 에지라면, 다음 에지에 대하여 break-simple( ) 실행한다.
만약 complex 에지라면, break-complex( ) 실행을 위하여 스택에 저장한다.

Break-complex( )
1. 중간 노드에 연결된 모든 에지를 찾는다.
2. 중복부분이 최대화되도록 에지들의 배열을 찾는다.
3. 필요하면 스타이너 점을 삽입하고 새로운 에지를 CAM에 저장한다.
4. 새로운 스타이너 노드를 break-simple( ) 실행을 위해 스택에 저장한다.

## Ⅳ. 실험결과

본 논문에서 제안된 알고리즘은 IBM PS2/70 컴퓨터에 내장된 연합처리기(4k 처리 소자)에서 수행되었다.

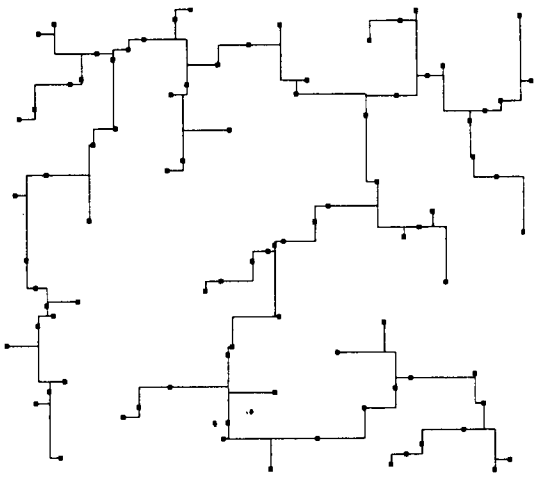

그림 9. 직교선형 스타이너 트리 구성 결과 (node 갯수: 90)
Fig. 9. Rectilinear Steiner Tree result. (The number of nodes : 90)

표 1. 직교선형 스타이너 트리 알고리즘의 결과
Table 1. Performance of the proposed algorithm.

| # of nodes | time(sec.) | RST/MST |
|---|---|---|
| 10 | 0.005 | 0.8412 |
| 20 | 0.013 | 0.8923 |
| 30 | 0.025 | 0.8634 |
| 40 | 0.036 | 0.8763 |
| 50 | 0.046 | 0.8622 |
| 60 | 0.061 | 0.8527 |
| 70 | 0.078 | 0.8679 |
| 80 | 0.096 | 0.8597 |
| 90 | 0.117 | 0.8749 |
| 100 | 0.141 | 0.8862 |

표1은 제안된 알고리즘의 결과를 나타낸다. 예제는 임의로 만들어졌고, 노드의 갯수를 10에서 100까지 변화시켰다. 직교선형 스타이너 트리와 MST 간의 비율은 0.84~0.87 정도로 산출되었고, 구성에 소요된 시간에 대한 결과는 매우 좋았다. 또한 소요된 시간은 문제의 크기가 커짐에 따라 선형적으로 증가하였다. 여러 가지 다른 종류의 예제를 풀어본 결과, 직교선형 스타이너 트리와 MST의 비율은 평균 0.88 정도로 나타났다. 구성된 직교선형 스타이너 트리와 MST간의 비율

의 상.하 한계치는 이론적으로 0.67≦ 직교선형 스타이너 트리/MST ≦1임이 알려져있다[11]. 그림9는 노드의 갯수가 90일 때 직교선형 스타이너 트리 구성결과이다.

연합처리기와 IBM PS2/70 컴퓨터 간의 통신은 마이크로 채널(Micro Channel)을 통해 이루어지기 때문에 수행 능률이 많이 낮아졌음에도 불구하고 실험결과는 매우 좋았다. 연합처리기와 주컴퓨터간의 통신방법이 개선된다면 보다 나은 결과를 얻을 수 있을 것이다.

## V. 결 론

본 논문에서는 CAM을 이용한 직교선형 스타이너 트리 구성에 관한 병렬알고리즘이 제안되었다. Z-형태 배열을 허용하기 위하여 노드 삽입 방법을 이용하였고, 결과적으로 초기 MST와 직교선형 스타이너 트리의 가중치 합의 비율을 향상시켰다. 제안된 알고리즘은 CAM의 병렬처리능력과 재배열의 불필요성 때문에 매우 고속으로 트리를 구성하였다. 문제의 크기가 증가함에 따라 트리 구성시간도 거의 선형적으로 증가되어 구성 알고리즘 복잡도에 커다란 향상을 보여주었다. 직교선형 스타이너 트리 문제는 그래프이론에서 등장하는 기초적인 문제이고, CAD와 VLSI 응용분야에서 자주 접하는 문제이므로, 본 논문에 제안된 CAM을 이용한 알고리즘을 통한 많은 응용이 가능하다.

## 참 고 문 헌


[1] F. K. Hwang, "On Steiner Minimal Trees with Rectilinear Distance", SIAM *Journal of Applied Mathematics*, vol.30, no.1, pp.104-114, 1976.

[2] M. R. Garay, D. S. Johnson, "The Rectilinear Steiner Tree Problem is NP-complete", SIAM *Journal of Applied* Mathematics, vol.32, no.4, pp.37-58, 1977.

[3] M. A. Breuer, "Min-cut Placement", Design Automation and Fault-Tolerant Computing, vol.1, pp.343-362, 1977.

[4] A. E. Dunlop, B. W. Kernighan, "A Procedure for Placement of Standard Cell VLSI Design", IEEE Transactions on Computer-Aided Design, vol.CAD-4, pp.92-98, 1985.

[5] K-W. Lee, C. Schen,"A New Global Router for Row-Based Layout", Proceedings of IEEE International Conference on Computer-Aided Design, 188, pp.180-183.

[6] F. K. Hwang, "An O(nlogn) Algorithm for Suboptimal Rectilinear Steiner Tree", IEEE Transactions on Circuits and Systems, CAS-26, pp.75-77, 1979.

[7] J. M. Ho, G. Vijayan, C. K. Wong, "New Algorithm for the Rectilinear Steiner Tree", IEEE Transactions on Computer-Aided Design of Integrated Circuits, CADIC-9, pp.185-193, 1990.

[8] K. Sakai, K. Tsuji, T. Matsumoto, "An Efficient Approximation Algorithm for the Steiner Tree Problem in Rectilinear Graphs", International Symposium on Circuits and Systems, pp.339-342, 1989.

[9] E. N. Gilbert, "Minimum Cost Communication Networks", Bell Sys. Tech. Journal, vol.9, pp.2209-2227, 1967.

[10] Coherent Processor Development System Version 2.2, User's Guide, Coherent Research Inc., 1990.

[11] T. G. Park, J. V. Oldfield, "Minimum Spanning Tree Generation with Content-Addressable Memory", IEE Electronics Letters, vol.29, no.11, pp.1037-1039, 1993.


### 저 자 소 개


朴 台 根(正會員) 제 32권 A편 4호 참조
현재 현대전자산업(주)
반도체 제2연구소 근무중.