論文99-36C-6-2

시간제약 조건을 고려한 CPLD 기술 매핑 알고리즘 개발

(Development of Technology Mapping Algorithm for CPLD by Considering Time Constraint)

金 熙 碩 *, 卞 相 畯 **

(Hi Seok-Kim and Sang Zoon-Byun)

요 약

본 논문에서는 시간제약 조건을 고려한 새로운 CPLD 기술매핑 알고리즘을 제안한다. 본 기술매핑 알고리즘은 주어진 논리식을 DAG로 구성하여 각 노드를 검색한 후, 출력에지가 2이상인 노드를 분할하지 않고 노드를 복제(replication)하여 DAG을 재구성함으로써 지연시간과 CLB의 개수가 최소화 되도록 하였다. 즉, 시간제약 조건과 소자의 지연시간을 이용하여 그래프 분할이 가능한 다단의 수를 정한다. 그런 다음 각 노드의 초기비용과 전체비용을 계산하여 CLB의 k-OR텀수보다 비용이 초과되는 노드를 분할하여 서브그래프를 구성한다.

분할된 서브그래프들은 collapsing을 통해 노드들를 병합하고 네 번째로 주어진 소자의 CLB안에 있는 k-OR텀 개수에 맞게 Bin packing를 수행하였다.

본 논문에서 제안한 기술매핑 알고리즘을 MCNC 논리합성 벤치마크 회로들에 적용하여 실험한 결과 기존의 기술 매핑 툴인 TEMPLA에 비해 CLB의 개수가 18% 감소되었다.

Abstract

In this paper, we propose a new technology mapping algorithm for CPLD under time constraint. In our technology mapping algorithm, a given logic equation is constructed as the DAG type, then the DAG is reconstructed by replicating the node that outdegree is more than or equal to 2. As a result, it makes delay time and the number of CLBs to be minimized. Also, after the number of multi-level is defined and cost of each nodes is calculated, the graph is partitioned in order to fit to k that is the number of OR term within CLB. The partitioned nodes are merged through collapsing and bin packing is performed in order to fit to the number of OR term within CLB.

In the results of experiments to MCNC circuits for logic synthesis benchmark, we can shows that proposed technology mapping algorithm reduces delay time and the number of CLBs much more than the existing tools of technology mapping algorithm.

* 正會員, 淸州大學校 電子·情報通信·半導體工學部
(School of Electronic, Semiconductor, Computer & Communication, Chongju University)

** 正會員, 大德大學 電子計算機科
(Department of Computer Technology, Taedok college)

接受日字:1999年3月20日, 수정완료일:1999年5月7日

I. 서 론

FPGA(Field-Programmable Gate Array)와 CPLD(Complexity Programmable Logic Device)는 설계시간이 짧고 비용(cost)이 적게 드는 장점으로 인하여 디지털 회로설계에 폭 넓게 사용되고 있다. 대부분의 FPGA는 LUT(Look-Up Table)를 기저로 한 논리블록이나 멀티플렉서를 기저로 한 논리블록들

로 구성되어 있다.

최근에는 조합논리회로나 순서논리회로를 k-입력을 갖는 LUT로 구현하는 FPGA 논리 합성 툴에 대한 연구가 활발히 진행되고 있다. LUT를 기저로 한 기술매핑(Technology Mapping)툴들의 최종목표는 LUT의 개수를 최소화하고 레벨을 줄여 소자의 면적과 지연시간을 최소화하는 성능개선에 촛점을 맞추고 있다.[3-6] 이에 반하여 CPLD는 PLA-style의 논리 블록들로 구성되어 있으며 Altera사에서 생산되는 MAX9000 계열의 경우, 내부를 구성하고 있는 매크로 셀(macrocell)은 그림 1과 같다.[8]

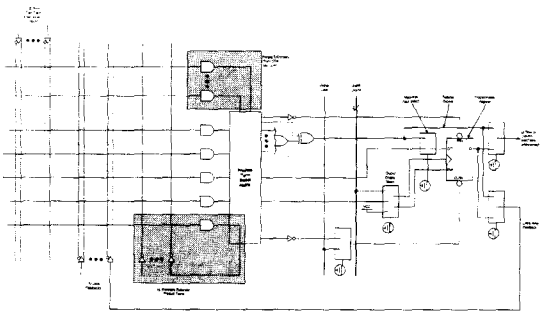


그림 1. MAX9000 계열의 매크로 셀
Fig. 1. Macrocell of MAX9000 series.

CPLD용 기술매핑 알고리즘에 관한 연구는 CAD 툴을 공급하는 Vendors들이 기술매핑 알고리즘을 독점하거나 거의 발표되지 않아 논리합성기술에 대한 연구는 매우 미진한 상태이다.[8]

따라서 본 논문에서는 주어진 시간제약(Time constraint)조건하에서 디지털 시스템을 CPLD로 구현할 수 있는 새로운 기술매핑 알고리즘(TMCPLD)을 제안한다.

제안한 기술매핑 알고리즘은 논리회로의 조합 논리회로 부분을 DAG(Directed Acyclic Graph)로 구성한 다음, 노드의 출력 에지(outdegree)의 개수가 2 이상인 경우에만 복제(replication)하여 DAG를 재구성함으로써 기존의 기술 매핑 툴인 TEMPLA가 노드의 출력 에지가 2 이상인 경우 분할하여 각각의 CLB (Control Logic Block)로 구현하는 것에 비해 지연시간과 CLB의 개수를 줄일수 있다.[8]

복제에 의해 재구성된 DAG는 OR텀의 개수를 각 노드의 비용으로 계산하여 CLB의 OR텀의 개수인 k-OR텀수보다 비용이 큰 노드를 분할점(cutpoint)으로 결정하고 이를 그래프 분할(Graph Partition)한 다음, Collapsing을 통해 부울식을 생성하고 이를 k-OR텀수에 맞게 Bin packing을 수행하게 된다. 또한, 본 논문에서 제안한 기술매핑 결과와 TEMPLA의 기술매핑 결과를 AMD사의 MACH 4와 Altera사의 MAX 계열(series)로 구현하여 MCNC 벤치마크 실험을 수행하였다.[1, 2]

Ⅱ. 관련연구

기술 매핑은 논리회로 자동설계의 한 과정으로써 논리식을 주어진 소자를 이용하여 기능적으로 동일한 회로를 구현하는 것이다. 일반적으로 논리 합성 툴들은 논리최소화를 거쳐 기술매핑을 수행하게 되는데, 기존에 제안된 FPGA 기술 매핑 방법들은 지연 시간의 최소화나 면적의 최소화에 중점을 두고 있다.

FPGA 기술 매핑 알고리즘인 Chortle-d, Mis-pga(delay), DAG-map, Flow-Map등은 구현된 회로의 성능을 향상하기 위하여 지연시간의 최소화에 중점을 두고 있으며 Chortle-crf, Vismap, Mis-pga (improved), X-map등은 면적의 최소화를 위하여 LUT 수의 최소화를 목적으로 제안되었다.

반면에 CPLD 기술 매핑 알고리즘으로는 최근에 발표된 TEMPLA가 있다.

TEMPLA는 Optimal Tree Mapping, Partial Collapsing, Bin Packing의 3단계로 나누어 기술 매핑을 수행한다. 즉, DAG에서 출력 에지의 개수가 2인 노드를 분할하여 서브그래프들의 집합을 형성하는 Optimal Tree Mapping과 분할된 서브그래프들을 CPLD의 내부를 형성하고 있는 PLA 블록의 AND-OR 2단 구조에 맞도록 부울식을 생성하는 Partial Collapsing과정, Partial collapsing에서 생성된 부울식을 PLA 블록에 집어넣기 위해 FFD(First Fit Decreasing) bin packing 알고리즘을 사용한 Bin Packing을 수행한다.

그러나 TEMPLA의 경우에 노드의 출력 에지의 개수가 2이상인 노드들을 모두 분할하여 팬 아웃 프리(fanout-free) 트리로 구성함으로써 알고리즘의 수행시간이 빠르다는 장점은 있으나 그래프 분할 후 생성된 부울식의 레벨수가 증가되어 주어진 시간 제약조건을 초과할 수 있으며 팬 아웃 프리트리를 분할한 만큼의 지연시간과 CLB의 개수가 증가되는 단점이 있다.

따라서 본 논문에서는 복제를 이용하여 팬 아웃 프

리 트리를 구성하고, 주어진 시간 제약 조건내에서 다단으로 그래프를 분할함으로써 주어진 시간제약 조건 안에서 회로의 지연 시간을 갖도록 구성하고 CLB의 개수도 최소화 할 수 있는 새로운 CPLD 기술 매핑 알고리즘을 제안한다.

Ⅲ. 기본 정의

먼저 다음의 용어들을 정의한다. 입력으로 사용되는 EDIF 네트리스트나 부울식은 DAG로 구성하여 G(N, E)로 표현된다. N은 DAG에서 노드(node)들의 집합이며, 각 노드는 AND, OR, NOT 게이트들 중의 하나의 기능만을 가지고 있다.[9]

E는 에지(edge)들의 집합이며 PI(primary input)에서 PO(primary output) 방향으로 신호 전달 경로를 나타낸다.

(정의 1) DAG를 구성하고 있는 노드들 중 출력 에지의 개수가 1인 노드를 Feasible 노드라 정의하고, CPLD를 구성하고 있는 CLB 1개로 구현할 수 있는 노드이다.

(정의 2) DAG를 구성하고 있는 노드들 중 출력 에지의 개수가 2 이상인 노드를 Nonfeasisble 노드로 정의한다.

그림 2에서 n_n700 노드는 출력 에지의 개수가 2 이므로 (정의 2)에 해당되는 nonfeasible 노드이며, 나머지 노드들은 출력 에지의 개수가 1 인 feasible 노드가 된다.

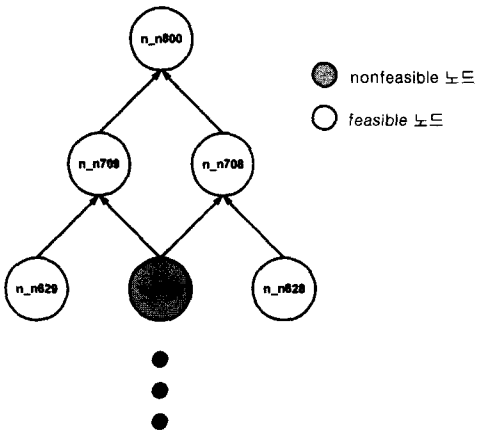


그림 2. Feasible 노드와 nonfeasible 노드
Fig. 2. Feasible node and nonfeasible node.

CPLD를 구성하고 있는 CLB 내의 OR 텀수를 k-OR텀수라 하고, 팬 아웃 프리 트리는 feasible 노드들만으로 구성된 DAG를 나타낸다. 팬 아웃 프리 트리를 구성하기 위해서는 DAG에서 각 노드를 검색하여 nonfeasible 노드가 있을 경우에는 nonfeasible 노드가 입력으로 쓰이는 노드들에 nonfeasible 노드의 하부 노드들까지 복사하는 복제의 방법을 사용한다.

복제에 의해 재구성된 DAG는 각 노드의 비용을 계산하여 그래프 분할을 수행하여야 한다. 이때 각 노드의 비용은 노드의 OR 텀수를 나타내며 노드의 입력이되는 다른 노드의 비용과는 무관하게 각 노드의 비용만을 초기 비용으로 계산한다.

전체 비용은 노드의 입력이되는 다른 노드의 비용을 고려하여 연산된 비용을 의미한다.

본 논문에서의 매핑 문제는 다음과 같이 정의한다.

입력이되는 EDIF 네트리스트나 부울식을 주어진 시간제약 조건 안에서 CLB의 수가 최소가 되도록 팬 아웃 프리 트리로 재구성된 그래프의 각 노드에 대해 비용을 계산하여 노드의 비용이 k-OR텀수를 초과할 경우 그래프 분할을 수행한다.

그래프의 비용 계산은 식 (1)과 같이 정의한다.

$$Cost(Node) = \alpha \cdot \Pi(Child_Node_Cost(Node)) + \beta \cdot \sum(Child_Node_Cost(Node)) \quad (1)$$

$Child_Node_Cost(Node)$: 계산하고자 하는 노드의 입력 노드들

식 (1)의 첫 번째 항은 계산하고자하는 노드의 현재 비용이 1인 경우로 AND 연산만을 가지고 있어 입력이 되는 노드의 비용들을 곱하므로서 비용을 계산할 수 있다. 두 번째 항은 계산하고자하는 노드의 현재 비용이 2 이상인 경우로 OR 연산만을 가지고 있어 입력이 되는 노드의 비용들을 합하므로서 비용을 계산할 수 있다. 비용 계산 식에서 α와 β는 수식 연산의 조건 값으로 계산하고자하는 노드의 현재 비용이 1인 경우는 α=1, β=0으로 주어지며, 계산하고자하는 노드의 현재 비용이 2이상인 경우는 α=0, β=1로 주어진다.

Ⅳ. CPLD를 위한 기술 매핑 알고리즘(TMCPLD)

본 논문에서 제안한 CPLD를 위한 기술매핑 알고

리즘은 크게 4가지 단계로 나눌 수 있다. 첫째, EDIF 네트리스트나, 부울식을 DAG로 구성하여 출력 에지의 개수가 2이상인 노드를 복제를 이용하여 DAG를 재구성하는 단계와 둘째, OR 텀의 개수를 각 노드의 비용으로 계산하는 초기비용과 팬 아웃 프리트리의 전체비용을 계산하여 비용이 초과하는 노드를 그래프 분할하는 단계, 셋째, 분할된 서브그래프의 노드를 병합하는 휴리스틱 collapsing 단계, 그리고 마지막으로 CLB내의 OR텀수에 맞추어 bin packing을 수행하는 단계 등으로 구성되어 있다.

1. DAG 생성

기술 매핑 툴들의 입력인 EDIF 네트리스트나 부울식은 그래프 형식인 DAG로 구성하게 되는데, DAG는 하나의 출력 변수에 대한 입력 변수들과 중간 출력을 노드로 형성한 팬 아웃 프리 트리로 나타낸다.

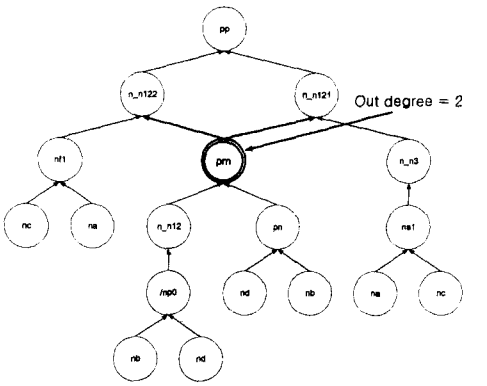
기존의 CPLD용 기술매핑 알고리즘인 TEMPLA은 팬 아웃 프리 트리를 구성하기 위해 출력 에지의 개수가 2개 이상인 노드들을 분할하여 새로운 서브그래프들을 만든다.

그러나 이러한 구성은 CPLD에 회로를 구현하였을 때 분리된 서브그래프만큼의 지연시간을 필요로 하게 되고 CLB의 개수도 늘어나게 되어 주어진 시간제약 조건내에서 회로를 구현하지 못하는 경우가 발생하게 된다.

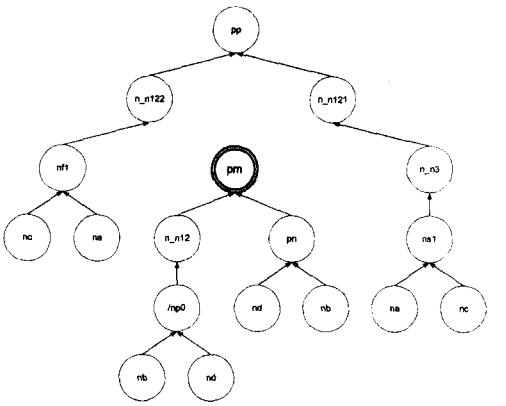
따라서 본 논문에서는 DAG에서 각 노드를 검색하여 출력 에지의 개수가 2이상인 노드는 분할하지 않고 복제하여 DAG를 재구성함으로써 지연시간과 CLB의 개수를 줄일 수 있는 팬 아웃 프리 트리를 구성한다.

그림 3은 출력 pp의 DAG에 대해 TEMPLA에서 구성한 팬 아웃 프리 트리와 본 논문에서 제안한 복제을 이용하여 재구성한 팬 아웃 프리 트리의 구성 예제이다.

그림 3의 (a)는 MCNC 벤치마크인 ALU2의 출력 중 하나인 pp를 DAG로 나타낸 그래프이고 (b)는 기존의 기술 매핑 툴인 TEMPLA에서 출력 에지의 개수가 2인 pm 노드를 서브그래프로 분리하여 2개의 팬 아웃 프리 트리를 2개의 CLB로 구현한 예이다. (c)는 본 논문에서 제안한 복제을 이용하여 DAG를 재구성한 팬 아웃 프리 트리로서 1개의 CLB로 구현이 가능하여 (b)보다 t만큼의 지연시간과 1개의 CLB의 개수가 감소하는 결과를 보였다.

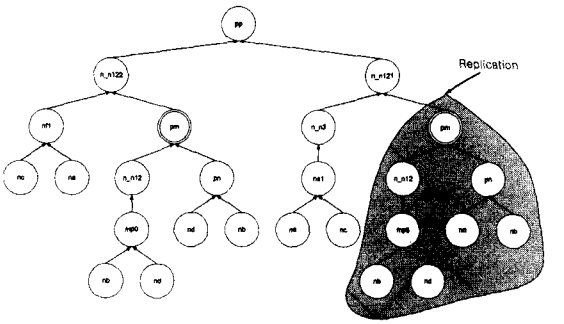


(a) DAG
(a) DAG



사용된 CLB 수 : 2개
전체 지연 시간 : 2t (t : 소자의 지연시간)

(b) TEMPLA의 팬 아웃 프리 트리
(b) Fanout-free tree of TEMPLA



사용된 CLB 수 : 1개
전체 지연 시간 : t

(c) TMCPLD의 팬 아웃 프리 트리
(c) Fanout free tree of TMCPLD

그림 3. 팬 아웃 프리 트리의 구성
Fig. 3. Construct of fanout free tree.

2. 그래프 분할(Graph Partition)

복제에 의해 재구성된 DAG를 주어진 시간 제약 조건이 허용되는 범위 안에서 최대한 다단으로 나누어야 한다. DAG를 다단으로 나누는 이유는 다단으로 분할할수록 사용되는 CLB의 개수가 줄어들기 때문이다.

따라서 본 논문에서는 시간제약 조건내에서 분할 가능한 다단의 수를 구하고 각노드의 초기비용과 전체 노드의 전체비용를 계산하여 분할조건 k-OR팀수를 초과하는 노드를 그래프 분할하게 된다.

1) 다단의 수

시간 제약의 조건 안에서 분할 가능한 다단의 수는 회로를 cascade로 구성할 경우의 구현 가능한 단수로서 시간 제약과 선택된 소자의 지연시간을 나누어서 구하며 식 (2)과 같다.

$$\text{다단의 수} = \frac{\text{시간 제약 조건}}{\text{소자의 지연 시간}} \qquad (2)$$

식 (2)에서의 다단의 수는 최소한 2이상의 값을 가져야 한다. 왜냐하면 다단의 수가 1인 경우는 하나의 출력에 대한 모든 부울식을 한 개의 CLB에 매핑하는 것은 불가능하기 때문에 다단의 수는 2 이상의 값을 가져야 한다.

2) 초기비용 계산

식 (2)에 의해 다단의 수가 결정되면 DAG에서 각 노드의 초기비용를 계산하게 되는데 PI노드에서 PO노드순으로 검색하여 OR팀이 있는 노드는 OR팀의 개수를 비용으로 계산하고 OR팀이 없는 경우에는 비용을 1로 계산한다. 초기비용을 계산하는 알고리즘은 그림 4와 같다.

```
Initial_Cost(node)
{
        Node_Cost = 0;
        NodeSet = 모든 node;
        While(NodeSet≠0)
        {
            Node_Cost = 각 node의 OR 팀 수;
        }
}
```

그림 4. 초기비용 계산 알고리즘
Fig. 4. Initial_Cost Algorithm.

3) 전체비용 계산

각 노드의 초기비용이 정해지면 팬 아웃 프리 트리의 전체 OR 텀의 개수인 전체비용을 다시 계산하게 되는데, 이때 전체비용계산을 위한 검색은 너비 우선 탐색(BFS) 방법을 사용하며 전체비용을 계산하는 알고리즘은 그림 5에 나타내었다.

```
Total_Cost(NodeSet, Node_Cost)
{
    While(NodeSet≠0)
    {
        if(Pre_Cost == 1) {
                α = 1, β = 0;
        }
        else if(Pre_Cost ≥ 2) {
                α = 0, β = 1;
        }
        Cost(Node) =
                α · Π(Child_Node_Cost(Node))
                + β · Σ(Child_Node_Cost(Node))
    }
}
```

그림 5. Total_Cost 알고리즘
Fig. 5. Total_Cost Algorithm.

4) 분할

팬 아웃 프리 트리의 전체비용이 계산되면 주어진 CLB의 OR 텀수에 맞게 그래프를 분할하여야 한다. 만약 계산된 전체비용이 주어진 CLB의 OR팀수를 초과할 경우에는 하나의 CLB로 회로를 구현할 수 없으므로 초과한 노드의 자식 노드들 중에 최대 값을 가진 노드를 분할한다. 그래프 분할 알고리즘은 그림 6과 같다.

그래프 분할 알고리즘은 계산된 전체비용을 이용하여 분할을 수행하게 되는데 식 (2)에서 구한 다단의 수의 범위 내에서 그래프를 분할한다. 그래프 분할의 첫 번째 단계는 어떤노드의 비용이 CLB내의 k-OR팀 수와 비교하여 클 경우에는 그 노드의 출력 에지를 분할하여 서브그래프로 만든다.

이때 k 값보다 큰 노드의 자식노드들을 검색하여 그중에 가장 큰 비용을 가지고 있는 노드가 있으면 이 노드를 우선 분할한다. 이렇게 분할하는 이유는 k 값

보다 큰 노드의 자식노드들은 모두 하나의 CLB로 구현 가능하기 때문에 가장 큰 값을 가진 노드들을 분할하여야만 CLB 개수를 줄일 수 있기 때문이다.

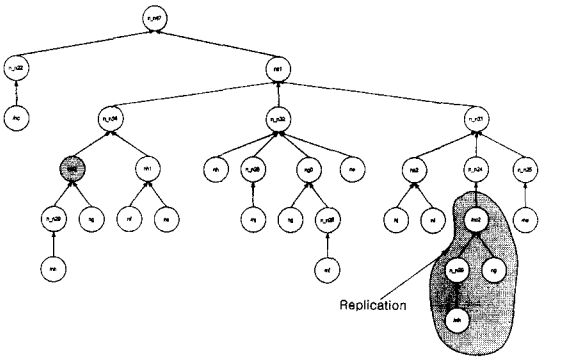
```
Graph_Partition(DAG, multilevel_number)
{
    Subgraph_Level = 0;
    k = 대상 소자에 대한 CLB 하나의 OR팀 수;

    Initial_Cost(DAG, NodeSet);

    while((multilevel_number-2)
                        ≥ Subgraph_Level)
    {
      while(k < node의 cost)
      {
          Total_Cost(DAG, NodeSet);
      }
      if(node_cost > k)
      {
       Separation_Subgraph(DAG, node_cost);
       Copy_Node(DAG, Sub_graph_po);
       Subgraph_Level ++;
      }
    }
}
```

그림 6. 그래프 분할 알고리즘
Fig. 6. Graph Partition Algorithm.

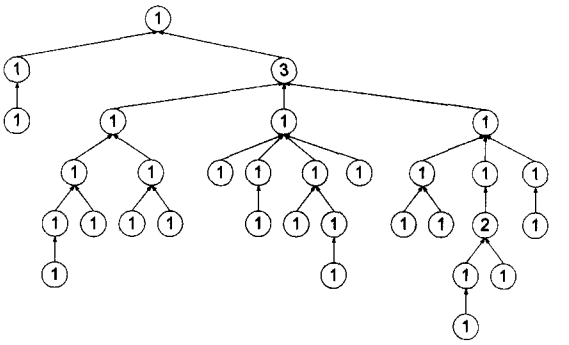
만약 분할할 자식 노드들의 비용이 모두 같을 경우에는 레벨의 깊이가 가장 깊은 자식노드를 우선 분할하며, 레벨의 깊이도 같은 경우는 좌측의 노드를 우선적으로 분할 한다. 이러한 분할은 Separation_Subgraph 알고리즘에 의해 수행되고, 분할이 끝나면 분할점에 새로운 노드을 생성하여 노드의 비용을 1로 주어지는데, 이 과정은 Copy_Node 알고리즘에 의해 수행된다. 노드분할은 식 (2)에서 구한 "다단의 수-2"만큼의 횟수로 수행한다. 그래프 분할 횟수를 "다단의 수-2"만큼 제한하는 이유는 그래프의 깊이(depth)가 깊어 그래프 분할을 수행했을 때 시간제약 조건을 초과하는 것을 방지하며, "다단의 수-2" 만큼의 그래프 분할 후 남아있는 그래프를 2단구조로 collapsing하여 CLB로 구현할 때 최소의 지연 시간을 가질 수 있도록 병렬 형태로 구성하면 2t의 지연시간이 필요로 하기 때문이다.

또한, 그래프 분할의 수행 횟수를 제한하기위해 분할한 횟수를 Subgraph_Level에 저장한다.

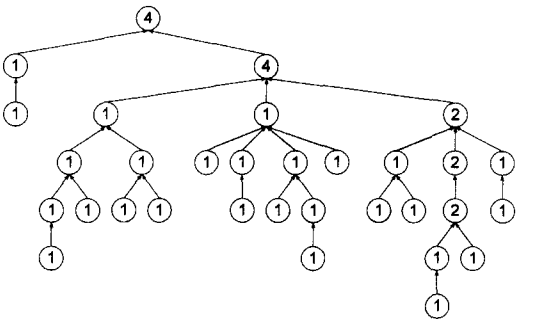
본 논문에서 제안한 그래프 분할 알고리즘을 이용한 예로 k=3이고 다단의 수가 2일때, MCNC 벤치마크 회로인 ALU4의 중간 노드 n_n47을 그래프 분할하는 과정을 그림 7에 나타내었다.



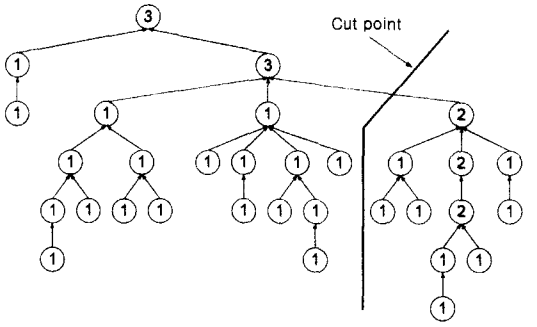
(a) DAG
(a) DAG



(b) 초기비용 계산
(b) Caculation of Initial cost



(c) 전체비용 계산
(c) Caculation of total cost



(d) 그래프 분할
(d) Graph Partition

그림 7. 그래프 분할
Fig. 7. Graph Partition.

그림 7의 (a)는 n_n47에 대한 DAG에서 오른쪽 노드 nc2를 왼쪽에 복제하여 생성한 그래프이고, (b)는 (a)를 Initial_Cost 알고리즘으로 각 노드의 초기비용을 계산한 그래프로 복제된 /nc2 노드의 초기비용이 2로 계산된 이유는 nc2 노드의 AND 연산이 인버터(Inverter)되어 OR 연산으로 변환되었기 때문이다. (c)는 Total_Cost 알고리즘을 이용하여 노드들의 전체비용을 계산한 결과이다. (d)는 조건 k=3일 때, nz1 노드의 비용이 4이므로 nz1의 자식 노드들인 n_n34, n_n32, n_n33 노드들 중 n_n33 노드의 비용이 가장 크므로 n_n33 노드를 분할하면 2개의 CLB로 매핑 가능하고 전체 지연시간은 2t의 지연시간을 가지게 된다.

3. Collapsing

그래프 분할 알고리즘에 의해 생성된 서브그래프는 CLB의 AND-OR로 구성된 2단 구조에 맞게 부울식을 추출하여야 한다.

그러기 위해서는 분할된 서브그래프의 최상위 노드를 새로운 출력으로 지정하여 부울식을 생성하는 collapsing을 수행하게 되는데, 본 논문에서는 SIS에서 제공되는 collapsing 알고리즘을 사용한다.[7]

```
n_n47 = /nc*nf*ne*/nh*ng + /nc*nh*/nj*ng
        */nf*ne + /nc*n_n33;
n_n33 = nj*nf*/ne*nh + nj*nf*/ne*/ng;
```

그림 8. Collapsing 결과
Fig. 8. Result of Collapsing.

그림 8은 그림 7에서 분할된 2개의 서브그래프를 collapsing하여 생성한 출력 n_n47과 n_n33에 대한 부울식이다.

4. Bin packing

FPGA용 기술 매핑 툴들에서 사용하는 Bin packing 알고리즘들은 LUT의 입력변수 제한에 맞추어 Bin packing을 수행한다.

그러나 CPLD의 경우 CLB의 내부가 입력부분이 AND-OR의 2단 구조로 구성되어 있어 입력변수의 개수와는 무관하기 때문에 출력단의 OR 팀수를 제한점으로 bin packing을 하게 된다.

따라서 본 논문에서는 collapsing 된 부울식을 CPLD의 CLB 내의 OR 팀수에 맞도록 Bin packing을 수행한다. Bin packing은 collapsing에 의해 생성된 부울식을 입력받아 우선 출력 변수들을 분리한다. 분리된 각각의 출력 변수에 대해 OR 텀 수를 계산하여 CLB의 OR 텀 수 k에 맞게 Bin packing을 수행하게 된다. 이때 시간 제약 조건을 고려하여 한 출력에 대해 2t의 지연시간을 갖도록 구성한다.

V. 실험결과

본 논문에서는 논리합성용 MCNC 벤치마크 회로들을 기존의 기술 매핑 툴들의 기술 매핑 결과와 본 논문에서 제안한 TMCPLD의 기술 매핑 결과를 비교하여 표 1에 나타내었다.

표 1에서는 TMCPLD의 기술매핑 결과가 DDMAP에 비해 약 22%의 CLB수가 감소됨을 알수 있고, TEMPLA에 비해 18% 감소되어 본 논문에서 제안한 기술매핑알고리즘의 결과가 우수함을 입증하였다.

그러나 수행시간은 TEMPLA와 마찬가지로 300MHz SUN SPARC 워크스테이션에서 수행한 결과, TMCPLD는 출력 에지의 개수가 2이상인 노드들을 복제하여 DAG을 재구성함으로써 노드의 수가 증가하여 TEMPLA에 비해 증가됨을 알수있다.

또한 표 2에서는 시간제약 조건하에서 다단의 수를 고려한 기술매핑 결과를 ALTERA사의 MAX-9000 계열로 구현하여 MAX_PLUS와 비교해보면 다단의 수가 5일 경우가 다단의 수가 3일 경우보다 8.9%의 감소율을 보여 다단으로 갈수록 많은 양의 CLB가 감소됨을 보였다.

표 1. 기존의 기술매핑툴들과 TMCPLD의 결과비교

Table 1. Result of comprison of the existing technology mapping and TMCPLD.

	MACH4 계열(AMD사)				
	DDMAP	TEMPLA		TMCPLD	
	CLB수	CLB수	수행시간 (sec)	CLB수	수행시간 (sec)
alu4	199	155	29.3	81	33
cps	159	120	18.3	119	22.1
apex4	193	193	30.2	139	50.6
misex3	214	154	27.8	147	32.8
ex5p	27	132	18.8	132	33.5
alu2	-	-	-	17	16.4
clip	-	-	-	11	4.4
duke	-	-	-	18	4.5
sao2	-	-	-	7	1.9
				DDMAP: 22%감소 TEMPLA: 18%감소	

표 2. 다단의 수를 고려한 기술매핑

Table 2. Technology mapping for multilevel number.

	MAX9000계열 다단의 수를 고려하지 않았을 경우 CLB 수	TMCPLD	
		MAX9000 계열(k=5)	
		다단의 수=3일 경우 CLB 수	다단의 수=5일 경우 CLB 수
alu4	249	213	203
cps	189	182	153
apex4	727	269	220
misex3	296	260	237
ex5pl	436	433	422
alu2	64	63	62
clip	43	42	40
duke	83	80	68
sao2	27	24	21

VI. 결 론

본 논문에서는 주어진 시간제약 조건을 고려한 새로운 CPLD 기술매핑 알고리즘(TMCPLD)을 제안하였다. 제안한 기술매핑 알고리즘는 주어진 논리식을 DAG로 구성한 후, 출력 에지의 개수가 2이상인 노드를 복제하여 DAG을 재구성함으로써 지연시간과 CLB의 개수가 최소화 되도록 하였다. 또한 시간제약 조건과 소자의 지연시간을 이용하여 분할 가능한 다단의 수를 정하고 각 노드의 초기비용과 전체비용을 계산하여 전체비용이 k-OR텀수 보다 큰 노드를 선택하여 그래프 분할한 다음, collapsing을 통해 분할된 노드들을 병합하여 부울식을 생성하고 이를 CLB내의 k-OR텀 개수에 맞게 Bin packing를 수행하게 된다.

실험결과 본 논문에서 제안한 새로운 CPLD 기술 매핑 알고리즘이 기존의 CPLD 기술 매핑 툴인 TEMPLA에 비해 CLB의 개수가 18% 감소함을 알 수 있다.

향후 연구 과제로서 본 논문에서 제안한 팬 아웃 프리 트리를 구성하기 위해 사용한 복제 방법에 제한을 두어 알고리즘의 전체 수행시간을 줄이는 연구와 순차 회로에 대한 기술 매핑 연구가 필요하다.

참 고 문 헌

[1] The Altera Data Book, Altera Corporation, 1996.

[2] The MACH 4 Family Data Sheet, Advanced Micro Devices, 1996.

[3] J. Cong and Y. Ding, "Simultaneous Depth and Area Minimization in LUT-based FPGA Mapping", UCLA Department of Computer Science Technical Report, CSD TR_9500001.

[4] A. H. Farrahi and m. Sarrafzadeh, "Complexity of the Lookup-Table Minimization Problem for FPGA Technology Mapping", IEEE Transactions and Systems, Vol. 13, No. 11, November 1994, pp. 1319 - 1332.

[5] R. J. Francis, J. Rose and Z. Vranesic, "Chortle-crf: Fast Technology Mapping for Lookup Table-Based FPGAs", 28th ACM/IEEE Design Automation Conference, June 1991, pp. 227-233.

[6] R. J. Francis, J. Rose and Z. Vranesic, "Technology Mapping of Lookup Table-Based FPGAs for Performance",

1991 IEEE Conference on Computer-Aided Design, pp. 568-571.

[7] E. M. Sentovice et al., "SIS: A System for Sequential Circuit Synthesis:, Technical Report UCB/ERL M92/ 41, Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1992.

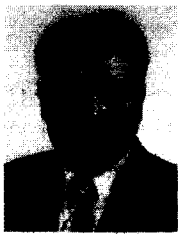
[8] J. H. Anderson and S. D. Brown, "Technolog Mapping for Large Complex PLDs", Proceeding of the 35th DAC, pp. 698-703, 1998.

[9] 김희석, 변상준 "EDIF Netlist를 이용한 PLD 설계용 툴 개발", 한국정보처리학회논문지, 제5권, 제4호, pp. 1025-1032

저 자 소 개

金 熙 碩(正會員)

1977년 한양대학교 전자공학과 졸업(공학사). 1980년 한양대학교 전자공학과(CAD 전공) 졸업(공학석사). 1985년 한양대학교 전자공학과(CAD 전공) 졸업(공학박사). 1987년 ~ 1988년 미국 Univ. of Colorado at Boulder 객원 교수. 1989년 ~ 1993년 청주대학교 전자공학과 부교수. 1996년 8월 ~ 1997년 7월 Univ. of California at Arvine 객원교수. 1993년 ~ 현재 청주대학교 전자·정보·반도체공학과 교수. 주관심분야는 CAD, 컴퓨터 아키텍춰, 컴퓨터 알고리즘



卞 相 畯(正會員)

1988년 청주대학교 전자공학과 졸업(공학사). 1990년 청주대학교 전자공학과(전자계산 및 계산기 전공) 졸업(공학석사). 1997년 청주대학교 전자공학과(전자계산 및 계산기 전공) 박사수료. 1993년~1996년 충남전문대학 전자계산기과 전임강사. 1997년 ~ 현재 대덕대의 전자계산기과 조교수. 주관심분야는 CAD, 컴퓨터 아키텍춰, 컴퓨터 알고리즘