

論文99-36T-12-10

시간제약 조건하에서 순차 회로를 위한 CPLD 기술 매핑 제어 알고리즘 개발

(Development of CPLD technology mapping control algorithm for Sequential Circuit under Time Constraint)

尹 忠 模 * , 金 才 鎮 **

(Chung-mo Youn and Jae-jin Kim)

요 약

시간제약 조건하에서 순차회로를 위한 새로운 CPLD(Complexity Programmable Logic Device) 기술 매핑 알고리즘을 제안한다. 본 기술매핑 알고리즘은 주어진 순차회로의 케환을 검출한 후 케환이 있는 변수를 임시 입력 변수로 분리한 후 조합논리 부분을 DAG(Directed Acyclic Graph)로 표현한다. DAG의 각 노드를 검색한 후, 출력 에지의 수가 2이상인 노드를 분할하지 않고 노드만을 복제(replication)하여 팬 아웃 프리 트리로 재구성한다. 이러한 구성 방법은 주어진 시간 조건 안에서 최소의 면적을 가질 수 있으며 처리 시간을 고려하기 위한 것이다. 기존의 CPLD 기술 매핑 알고리즘인 TEMPLA의 경우 팬 아웃 프리 트리를 구성 할 때 출력 에지의 수가 2이상인 노드를 서브 그래프로 분할함으로써 매핑 결과 시간 제약 조건을 초과할 수 있다. 또한, TMCPLD(Technology Mapping for CPLD)의 경우는 출력 에지의 수가 2 이상인 노드를 포함한 트리를 복제하여 전체의 노드수가 증가되어 전체 수행시간이 길어지는 단점을 가지고 있다. 이러한 단점을 보완하기 위해 노드만을 복제한 팬 아웃 프리 트리의 구성방법을 제안한다.

시간제약 조건과 소자의 지연시간을 이용하여 그래프 분할이 가능한 단단의 수를 정하고, 각 노드의 OR 텀수를 비용으로 하는 초기비용과 노드 병합 후 생성될 OR 텀수인 전체비용을 계산하여 CPLD를 구성하고 있는 CLB(Configurable Logic Block)의 OR텀수보다 비용이 초과되는 노드를 분할하여 서브그래프를 구성한다.

분할된 서브그래프들은 collapsing을 통해 노드들을 병합하고, 주어진 소자의 CLB안에 있는 OR텀 개수에 맞게 Bin packing을 수행하였다.

제안한 기술매핑 알고리즘을 MCNC 논리합성 벤치마크 회로들에 적용하여 실험한 결과 기존의 CPLD 기술 매핑 틀인 TEMPLA에 비해 CLB의 수가 15.58% 감소되었다.

Abstract

We propose a new CPLD(Complexity Programmable Logic Device) technology mapping algorithm improving run-time under time constraint. In our technology mapping algorithm, a given logic equation is constructed as the DAG(Directed Acyclic Graph) 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, run-time 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(Configurable Logic Block).

In the results of experiments to MCNC circuits for logic synthesis benchmark, we can show that proposed technology mapping algorithm reduces run-time and the number of CLBs much more than the TEMPLA.

* 正會員, 瑞逸大學 情報通信科

(Dept. of Information and Communication, Seoul College)

** 正會員, 清州大學校 電子工學科

(Dept. of Elec. Engineering, Chongju University)

接受日字:1999年10月20日, 수정완료일:1999年11月22日

I. 서론

FPGA(Field Programmable Gate Array)와 CPLD(Complexity Programmable Logic Device)는 설계시간이 짧고 비용(cost)이 적게드는 장점으로 인하여 디지털 회로설계에 폭 넓게 사용되고 있다. 대부분의 FPGA 내부는 LUT(Look-Up Table)를 기저로 한 논리블록이나 멀티플렉서를 기저로 한 논리블록들로 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 초점을 맞추고 있다.^{[3][4][5][6]} 이에 반하여 CPLD는 PLA-style의 논리 블록들로 구성되어 있다.

그림 1은 CPLD의 한 예로 Altera사에서 생산되는 MAX9000 계열의 내부를 구성하고 있는 매크로 셀(macrocell)의 내부를 나타내었다.

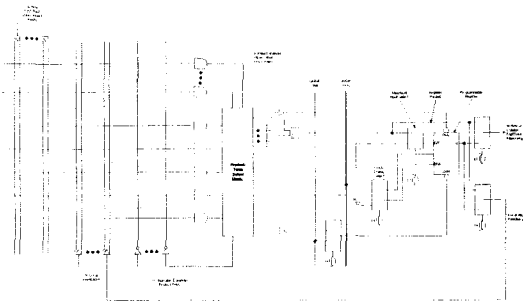


그림 1. MAX9000 계열의 매크로 셀
Fig. 1 Macrocell of MAX9000 series

PLA-style로 구성된 CPLD용 기술매핑 알고리즘에 관한 연구는 CAD툴을 공급하는 Vendors들이 기술매핑 알고리즘을 독점하거나 거의 발표되지 않아 논리합성기술에 대한 연구는 매우 미진한 상태이다.^[6]

또한 주어진 시간제약(Time constraint)조건하에서 디지털 시스템을 CPLD로 구현할 수 있는 기술매핑 알고리즘으로는 TMCPLD(Technology mapping for CPLD)만이 제안되었다.

그러나 TMCPLD는 조합 논리 회로에만 국한되어 있

어 주어진 시간 제약 조건안에서 순차회로를 위한 CPLD 기술 매핑 알고리즘은 제안되지 않았다. 또한, TMCPLD는 주어진 EDIF 네트리스트나 부울식을 DAG(Directed Acyclic Graph)로 구성한 후 노드의 출력 에지의 개수(outdegree)가 2 이상인 노드가 존재할 경우 노드를 포함한 하부의 노드들, 즉 트리를 전체 복사하여 팬 아웃 프리 트리(fanout free tree)를 구성하였다. 이러한 구성은 노드의 수가 증가되어 전체 수행시간이 길어지는 단점이 있다.

따라서 이러한 단점을 보완한 주어진 시간조건하에서 순차회로를 위한 CPLD 기술 매핑 알고리즘을 제안하였다.

제안한 기술매핑 알고리즘은 궤환이 있는 변수를 임시 입력 변수로 변환하여 궤환을 제거한 후 논리회로를 DAG로 구성한다. DAG는 각 노드를 검색하여 출력 에지의 개수가 2 이상인 노드가 있을 경우에 그 노드만을 복제(replication)하여 팬 아웃 트리로 DAG를 재구성함으로써 주어진 시간 조건에 맞도록 기술 매핑을 수행함으로써 기존의 기술 매핑 툴인 TEMPLA보다 적은 면적으로 구현할 수 있다.^[6]

복제에 의해 생성된 팬 아웃 프리 트리는 각 노드의 비용을 OR팀의 개수로 계산하여 구현 대상 소자를 구성하고 있는 CLB의 OR팀 개수인 k-OR팀수보다 비용이 큰 노드를 분할점(cutpoint)으로 결정하여 그래프 분할(Graph Partition)을 수행한다. 분할된 그래프는 Collapsing을 통해 2단 구조(two-level)의 부울식을 생성한다. 생성된 부울식의 OR 팀수가 k-OR팀수보다 많은 경우 k-OR팀수에 맞게 Bin packing을 수행하게 된다. 본 논문에서 제안한 기술매핑 알고리즘과 기존의 기술 매핑 알고리즘인 DDMAP[9], TEMPLA[6]의 기술매핑 적용 결과를 비교하기 위해 MCNC 벤치마크 회로들을 AMD사의 MACH 4로 구현하여 비교하였다.^{[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-Map 등은 구현된 회로의 성능을 향상하기 위하여 지연시간의 최소화에 중점을 두고 있으며 Chortle-crf, Vismap, Mis-pga(improved), X-map 등은 면적의 최소화를 위하여 LUT 수의 최소화를 목적으로 제안되었다.

반면에 CPLD 기술 매핑 알고리즘은 거의 제안되지 않았으며 최근에 제안된 알고리즘으로는 TEMPLA, TMCPLD 등이 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 트리 트리로 구성함으로써 알고리즘의 수행시간이 빠르다는 장점은 있으나 그래프 분할 후 생성된 부울식의 레벨 수가 증가되어 주어진 시간 제약조건을 초과할 수 있으며 팬 아웃 프리트리를 분할한 만큼의 지연시간과 CLB의 개수가 증가되는 단점이 있다.

TMCPLD의 경우는 복제를 이용하여 팬 아웃 프리트리를 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 알고리즘이다.

그러나 TMCPLD는 팬 아웃 프리 트리 생성시 출력 에지의 개수가 2이상인 노드를 포함한 트리를 복제함으로써 노드의 수가 증가하여 전체 수행시간이 길어지는 단점이 있다.

III. 시간제약 조건하에서 순차 회로를 위한 CPLD 기술 매핑 알고리즘 개발

본 논문에서 제안한 기술매핑 알고리즘은 크게 4가지 단계로 나눌 수 있다. 첫 번째 단계로 EDIF 네트리스트나 부울식에서 케환을 검출하여 케환이 되는 변수를 임시 입력 변수로 재구성하여 플립-플롭을 제외한 조합논리 부분을 DAG로 구성한 후 각 노드를 검색하여 출력 에지의 개수가 2이상인 노드만을 복제하여 팬 아웃 프리 트리를 구성하는 트리 매핑(tree mapping) 단계이다. 두 번째 단계로 OR 텀의 개수를 각 노드의 비용으로 계산하는 초기비용과 팬 아웃 프리 트리의 전체비용을 계산하여 비용이 초과하는 노드를 그래프 분할하는 그래프 분할 단계, 셋째 단계로 분할된 서브그래프의 노드를 병합하는 휴리스틱 collapsing 단계, 그리고 마지막으로 CLB내의 OR텀수에 맞추어 bin packing을 수행하는 단계 등으로 구성되어 있다. 우선 본 논문의 알고리즘을 정의하기 위한 기본 정의에 대해 설명한다.

1. 기본 정의

먼저 다음의 용어들을 정의한다. 입력으로 사용되는 EDIF 네트리스트나 부울식은 DAG로 구성하여 $G(N, E)$ 로 표현된다. N 은 DAG에서 노드(node)들의 집합이며, 각 노드는 AND, OR, NOT 게이트들 중의 하나의 기능만을 가지고 있다.^[10]

E 는 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 Nonfeasible 노드로 정의한다.

CPLD를 구성하고 있는 CLB 내의 OR 텀수를 k-OR 텀수라 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 최상위

노드만을 복사하는 복제의 방법을 사용한다.

(정의 3) 하나의 출력 변수에 대한 부울식의 입력단에 출력 변수가 존재하면 이 출력변수는 궤환(feedback)되었다고 정의한다

(정의 4) 궤환된 출력 변수에 대한 입력단의 출력변수는 임시 입력 변수 (TIV : Temporary Input Variable)라 정의하고, 변수의 명칭을 임시 수정한다.

임시 입력 변수는 collapsing 수행 후 초기의 출력 변수로 환원한다.

이러한 복제를 하기 위해서는 궤환을 미리 검출하여 궤환을 없애야 한다. 노드 검색에 의해 검출된 궤환은 임시 입력 변수로 변환하여 궤환을 없앤 후 복제가 이용하여 팬 아웃 프리 트리를 구성하고 각 노드의 비용을 계산하게 된다. 각 노드가 가지고 있는 부울식의 OR 텀수를 초기 비용으로 정의하며, 입력이되는 노드의 비용을 고려하여 노드 병합 후의 OR 텀수를 전체 비용이라 정의한다.

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 곱하므로써 비용을 계산할 수 있다. 두 번째 항은 계산하고자하는 노드의 현재 비용이 2 이상인 경우로 OR 연산만을 가지고 있어 입력이 되는 노드의 비용들을 합하므로써 비용을 계산할 수 있다. 비용 계산 식에서 α 와 β 는 수식 연산의 조건 값으로 계산하고자하는 노드의 현재 비용이 1인 경우는 $\alpha=1$, $\beta=0$ 으로 주어지며, 계산하고자하는 노드의

현재 비용이 2이상인 경우는 $\alpha=0$, $\beta=1$ 로 주어진다.

2. 트리 매핑

기술 매핑 툴들의 입력인 EDIF 네트리스트나 부울식은 그래프 형식인 DAG로 구성하게 되며 DAG는 하나의 출력 변수에 대한 입력 변수들과 중간 출력을 노드로 하여 feasible 노드만으로 구성된 팬 아웃 프리 트리로 재구성된다.

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.

이러한 단점을 보완하기 위해 TMCPLD에서는 DAG에서 각 노드를 검색하여 출력 에지의 개수가 2이상인 노드는 분할하지 않고 노드를 포함한 트리를 복제하여 DAG를 재구성함으로써 시간 제약 조건안에서 회로를 구현할 수 있는 팬 아웃 프리 트리를 구성하였다.

그러나 TMCPLD의 경우 트리를 복제하므로 노드의 수가 증가되어 전체 수행 시간이 길어지는 단점이 있다. 또한 TMCPLD는 조합논리 회로만을 고려하였으며 궤환이 있는 순차회로는 고려하지 않았다.

따라서 본 논문에서는 TMCPLD의 단점인 전체 수행 시간을 줄이고 궤환이 있는 순차회로에도 적용할 수 있는 새로운 CPLD 기술 매핑 알고리즘을 개발하였다. 본 논문의 팬 아웃 프리 트리 구성은 각 노드를 검색하여 궤환이 있을 경우 궤환된 출력 변수를 임시 입력 변수로 선정된 후 궤환을 없앤다. 궤환을 제거한 그래프는 재검색을 통해 출력 에지의 개수가 2이상인 노드는 분할하지 않고 노드만을 복제하여 팬 아웃 프리 트리를 생성한다.

궤환의 검출은 그림 2의 궤환 검출 알고리즘을 이용하였다.

그림 3은 그림 2의 궤환 검출 알고리즘을 적용하여 궤환을 검출한 후 궤환을 제거하고 각 노드를 재검색하여 출력에지의 수가 2이상인 노드들만을 복제하여 구성한 팬 아웃 프리 트리의 구성 예이다. 그림 3의 회로는 순차회로의 예로 4비트 up/down 카운터의 출력들

```

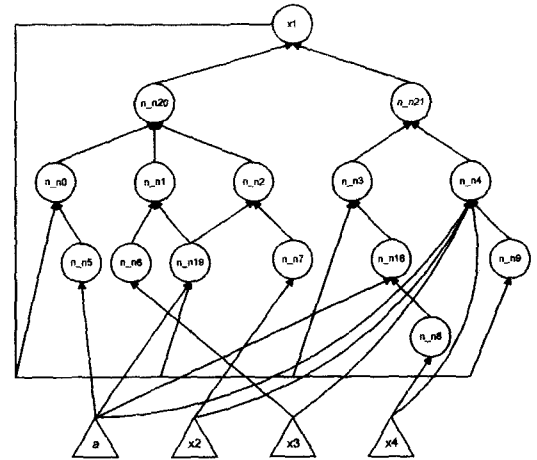
Ext_Feedback_Node(DAG)
{
    while(Pri_Output() = 0)
        // Pri_Output(): 최종 출력들의 집합
        {
            OutNode = OutN(DAG); // 최종 출력 노드
            // 를 하나 입력
            VisitN(OutNode); // OutNode를 방문
        }
}

VisitN(ParentNode)
{
    ID = 1;
    while(NodeSet = 0) // NodeSet : 노드들의 집합
        while(tempNode = ChildN(ParentNode))
            // 자식 노드들을 모두 방문할 때 까지
            {
                ID = ID + 1;
                if(tempNode ≠ Visited_Nodes)
                    // Visited_Nodes : 이미 방문한 노드들
                    {
                        tempNode_id = ID;
                        putN(tempNode);
                    }
                elseif(tempNode = Visited_Nodes)
                    {
                        if(tempNode_id ≥ ParentNode_id)
                            FeedbackN = tempNode;
                            // 퀘환 노드 집합에
                            tempNode 추가
                    }
            }
    ParentNode = get();
}
    
```

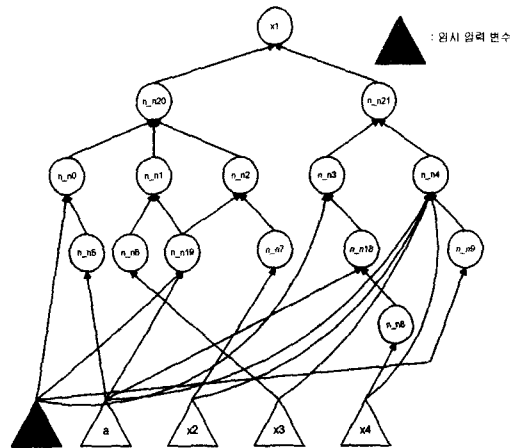
그림 2. 퀘환 검출 알고리즘
Fig. 2 Feedback detection Algorithm.

중에서 x1을 DAG로 구성하는 방법과 생성된 DAG에 대해 TEMPLA과 TMCPLD에서 구성한 팬 아웃 프리

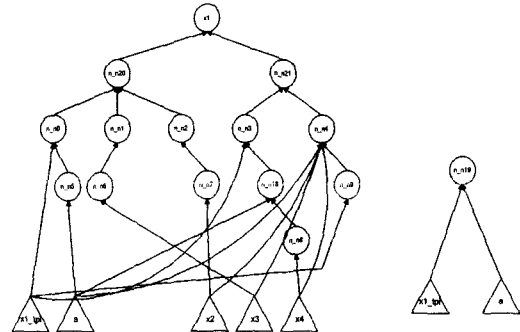
트리와 본 논문에서 제안한 노드만을 복제하여 재구성한 팬 아웃 프리 트리의 구성 예제이다.



(a) 그래프 구성
(a) Construction Graph

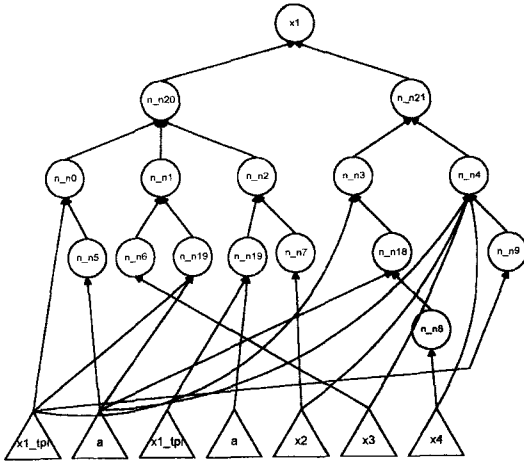


(b) 퀘환을 제거한 후 구성된 DAG
(b) Constructed DAG after removed of feedback

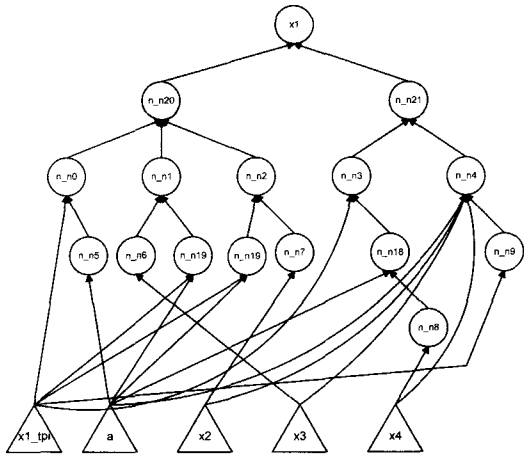


(c) TEMPLA의 팬 아웃 프리 트리
(c) Fanout-free tree of TEMPLA

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



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



(e) 본 논문의 팬 아웃 프리 트리
(e) Fanout free tree of in this paper

그림 3. (계속)
Fig. 3 (Continued)

그림 3의 (a)는 4비트 up/down 카운터의 출력들 중 하나인 x1에 대한 그래프 구성이며 (b)는 궤환을 검출한 후 궤환을 제거하여 생성된 DAG이다. (c)는 TEMPLA의 팬 아웃 프리 트리를 구성하는 방법으로 출력 에지의 개수가 2이상인 n_n19 노드를 서브그래프로 분리하여 2개의 그래프로 나타낸 팬 아웃 프리 트리 구성의 예이고, (d)는 TMCPLD에서 제안한 복제를 이용하여 DAG를 재구성한 팬 아웃 프리 트리로서 출력 에지의 수가 2 이상인 노드를 포함한 트리를 복제하기 때문에 노드의 수가 증가되어 처리 시간이 길어지는 단점이 있다. (e)는 본 논문에서 제안한 노드만을

복제하여 구성한 팬 아웃 프리 트리로서 TMCPLD의 단점을 개선하기 위해 출력 에지의 수가 2이상인 노드가 존재할 때 노드를 포함한 트리를 복제하는 TMCPLD와는 달리 그 노드만을 복제함으로써 복제에 의한 노드의 수가 증가되는 것을 최대로 억제하여 전체 수행시간이 줄어들게 된다.

3. 그래프 분할(Graph Partition)

복제에 의해 재구성된 DAG를 주어진 시간 제약 조건이 허용되는 범위 안에서 최대한 다단으로 나누어 야 한다. DAG를 다단으로 나누는 이유는 다단으로 분할 할수록 사용되는 CLB의 개수가 줄어들기 때문이다. 따라서 본 논문에서는 시간제약 조건내에서 분할 가능한 다단의 수를 구하고 각노드의 초기비용과 전체노드의 전체비용을 계산하여 분할조건 k-OR텀수를 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 클럭 주기를 나누어서 구하며 식 (2)과 같다.

$$\text{다단의 수} = \frac{\text{시간 제약 조건}}{\text{소자의 클럭 주기}} \quad (2)$$

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) 비용 계산

식 (2)에 의해 다단의 수가 결정되면 DAG에서 각 노드의 초기비용을 계산하게 되는데 PI노드에서 PO노드순으로 검색하여 OR텀이 있는 노드는 OR텀의 개수를 비용으로 계산하고 OR텀이 없는 경우에는 비용을 1로 계산한다. 각 노드의 초기비용이 정해지면 팬 아웃 프리 트리의 전체 OR 텀의 개수인 전체비용을 다시 계산하게 되는데, 이때 전체비용계산을 위한 검색은 입력이 되는 모든 노드의 초기 비용을 고려하여 노드 병합시 생성되는 OR 텀수를 계산해야 하므로 그래프 검색 알고리즘들 중에서 모든 연관된 노드를 검색하면서 진행되는 너비 우선 탐색(BFS) 방법을 사용하여 비용을 계산한다. 비용 계산을 위한 알고리즘은 그림 4에 나타내었다.

```

Initial_Cost(node)
{
    Node_Cost = 0;
    NodeSet = 모든 node;
    While(NodeSet≠0)
    {
        Node_Cost = 각 node의 OR 텀 수;
    }
}
Total_Cost(NodeSet, Node_Cost, Re_NodeSet)
{
    While(NodeSet≠0)
    {
        if(Pre_Cost == 1)//Pre_Cost : 현재 노드의
        비용
        {
            α = 1, β = 0;
        }
        else if(Pre_Cost ≥ 2) {
            α = 0, β = 1;
        }
        Cost(Node) = α · Π (hild_Node_Cost
        (Node)) + β · ∑(Child_Node
        _Cost(Node))
        // Child_Node_Cost : 비용 계산을 위
        한 현재 노드의 입력의 되는 모든
        자식 노드
    }
    if(NodeSet == Re_NodeSet)
    // Re_NodeSet : 복제에 의해 재구성된 팬 아웃
    프리 트리에서 복제된 노드들
        Re_NodeSet_Cost = NodeSet_Cost;
}
    
```

그림 4. 비용 계산 알고리즘
Fig. 4 Cost Calculation Algorithm.

3) 그래프 분할 알고리즘
팬 아웃 프리 트리의 전체비용이 계산되면 주어진 CLB의 OR 텀수에 맞게 그래프를 분할하여야 한다. 만약 계산된 전체비용이 주어진 CLB의 OR텀수를 초과할 경우에는 하나의 CLB로 회로를 구현할 수 없으므로

초과한 노드의 자식 노드들 중에 최대 값을 가진 노드를 분할한다. 그래프 분할 알고리즘은 그림 5와 같다.

```

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

    Initial_Cost(DAG, NodeSet);

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

그림 5. 그래프 분할 알고리즘
Fig. 5 Graph Partition Algorithm

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-OR텀수와 비교하여 클 경우에는 그 노드의 출력 에지를 분할하여 서브그래프로 만든다.

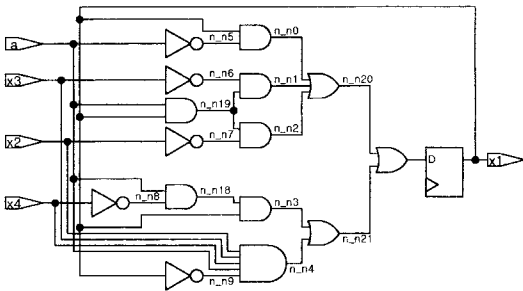
이때 k 값보다 큰 노드의 자식노드들을 검색하여 그 중에 가장 큰 비용을 가지고 있는 노드가 있으면 이 노드를 우선 분할한다. 이렇게 분할하는 이유는 k 값보다 큰 노드의 자식노드들은 모두 하나의 CLB로 구현 가능하기 때문에 가장 큰 값을 가진 노드들을 분할하여야만 CLB 개수를 줄일 수 있기 때문이다.

만약 분할할 자식 노드들의 비용이 모두 같을 경우에는 레벨의 깊이가 가장 깊은 자식노드를 우선 분할하며, 레벨의 깊이도 같은 경우는 좌측의 노드를 우선

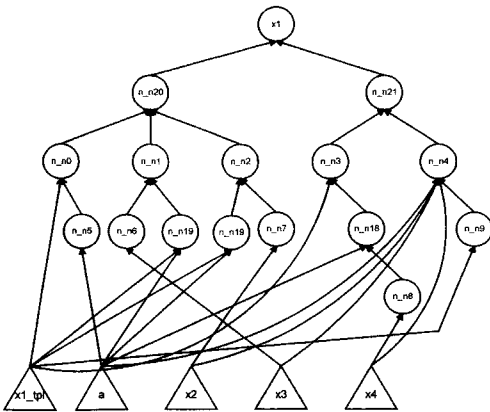
적으로 분할한다. 이러한 분할은 Separation_Subgraph 알고리즘에 의해 수행되고, 분할이 끝나면 분할점에 새로운 노드를 생성하여 노드의 비용을 1로 주어지는데, 이 과정은 Copy_Node 알고리즘에 의해 수행된다. 노드 분할은 식 (2)에서 구한 “다단의 수-2”만큼의 횟수로 수행한다. 그래프 분할 횟수를 “다단의 수-2”만큼 제한하는 이유는 그래프의 깊이(depth)가 깊어 그래프 분할을 수행했을 때 시간제약 조건을 초과하는 것을 방지하며, “다단의 수-2”만큼의 그래프 분할 후 남아있는 그래프를 2단구조로 collapsing하여 CLB로 구현할 때 최소의 지연 시간을 가질 수 있도록 병렬 형태로 구성하면 $2t$ 의 지연시간이 필요로 하기 때문이다.

또한, 그래프 분할의 수행 횟수를 제한하기 위해 분할한 횟수를 Subgraph_Level에 저장한다.

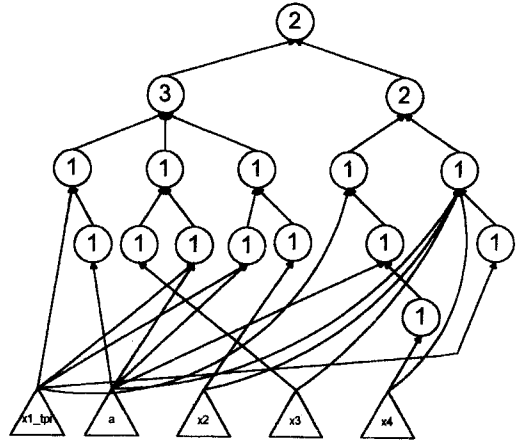
본 논문에서 제안한 그래프 분할 알고리즘을 이용하면 예로 $k=3$ 이고 다단의 수가 2일때, 4비트 up/down 카운터의 출력들 중 하나인 $x1$ 에 대한 그래프 분할 과정을 그림 6에 나타내었다.



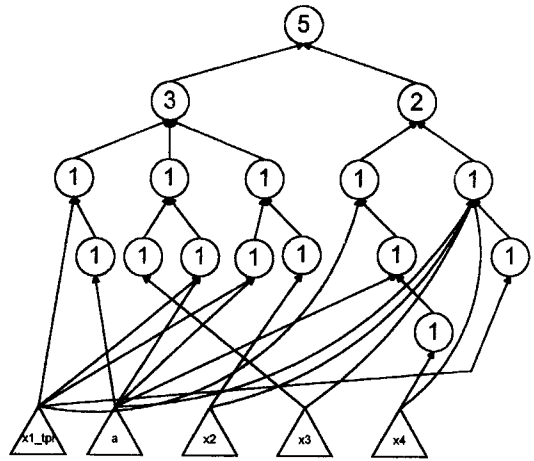
(a) 회로도
(a) Circuit



(b) 팬 아웃 프리 트리
(b) Fanout free tree

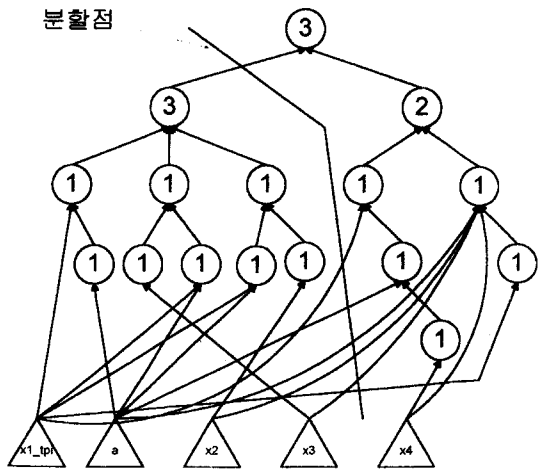


(c) 초기비용 계산
(c) Caculation of Initial cost

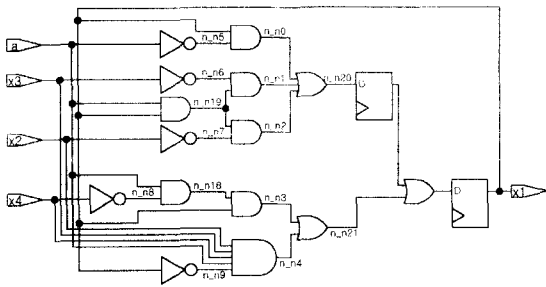


(d) 전체비용 계산
(d) Caculation of total cost

분할점



(e) 그래프 분할
(e) Graph Partition



(f) 그래프 분할 후의 회로도
(f) Circuit after graph partition

그림 6. 그래프 분할
Fig. 6 Graph Partition

그림 6의 (a)는 4비트 up/down 카운터의 출력들중의 하나인 x1에 대한 회로도이고 (b)는 (a)의 회로도를 DAG로 표현한 후 노드를 복제하여 생성한 팬 아웃 트리 트리이다. (c)는 (b)를 Initial_Cost 알고리즘으로 각 노드의 초기비용을 계산한 그래프이고 (d)는 Total_Cost 알고리즘을 이용하여 노드들의 전체비용을 계산한 결과이다. (e)는 조건 k=3일 때, 출력 x1의 비용이 5 이므로 x1의 자식 노드들중에서 비용이 가장 큰 n_n20 노드를 분할한 그래프 분할의 결과를 나타낸 그림이다. (f)는 그래프 분할 후의 회로도를 나타낸 그림이다. 따라서 출력 x1을 CPLD로 구현했을 경우 2개의 CLB로 구현되며 소자의 지연시간을 t라고 했을 때 전체 지연시간은 2t가 된다.

4. 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.⁽⁶⁾

그림 7은 그림 6에서 분할된 2개의 서브그래프를 collapsing하여 생성한 출력 n_n20과 x1에 대한 부울식이다.

$$x1 = n_n20 + x1*/x4*a + /x1*x2*x3*x4*a;$$

$$n_n20 = x1*/a + x1*/x3*a + x1*/x2*a;$$

그림 7. Collapsing 결과
Fig. 7. Result of Collapsing.

5. 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 각각의 출력에 대한 OR 텀수를 제한점으로 bin packing을 수행 하게 된다.

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

III. 실험결과

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

표 1. 기존의 기술매핑툴들과 본 논문의 기술 매핑 알고리즘과의 결과비교

Table 1. Result of comprison of the existing technology mapping and technology mapping algorithm of in this paper.

| | MACH4 계열(AMD사) | | | | |
|-----------|----------------|--------|-----------|--------------------|--------------------------------|
| | DDMAP | TEMPLA | | 본 논문에서 제안한 알고리즘 | |
| | CLB수 | CLB수 | 수행시간(sec) | CLB수 | 수행시간(sec) |
| s38417 | 1208 | 603 | 495.2 | 479 | 502.7 |
| seq | 337 | 229 | 52.8 | 219 | 54.1 |
| fir | 1424 | 249 | 123.3 | 199 | 128.1 |
| fsm8_8_13 | 58 | 49 | 5.5 | 49 | 5.8 |
| pmac | 911 | 237 | 126.6 | 232 | 131.3 |
| psdes | 301 | 151 | 37 | 119 | 39.4 |
| sort | 275 | 138 | 29.8 | 101 | 31.2 |
| | | | | DDMAP: 69.03%감소 | TEMPLA: 4.64%증가 15.58%감소 |

수행시간을 비교하기 위해 TEMPLA과 같은 조건인 300MHz SUN SPARC 워크스테이션에서 수행하였다. 또한 대상 소자도 같은 MACH4를 선정하여 알고리즘을 적용한 결과를 비교하였다.

표 1에서는 본 논문에서 제안한 기술매핑 결과 수행시간은 TEMPLA보다 4.64% 증가되었으나 면적에서는 DDMAP에 비해 약 69.08%의 CLB수가 감소됨을 알수 있고, TEMPLA에 비해 15.58% 감소된 결과를 보였다.

그림 8은 기존의 기술 매핑 알고리즘과 본 논문에서 제안한 알고리즘과의 면적을 비교한 그래프이다.

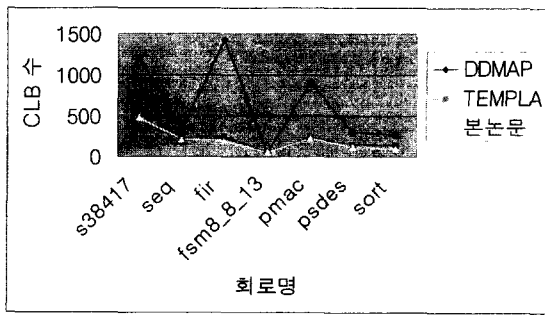


그림 8. 면적에 대한 비교
Fig. 8. Comprison for area.

IV. 결 론

본 논문에서는 주어진 시간제약 조건하에서 순차회로를 위한 새로운 CPLD 기술매핑 알고리즘을 제안하였다. 제안한 기술매핑 알고리즘은 EDIF 네트리스트나 논리식을 검색하여 케환을 검출하고, 검출된 케환을 임시 입력 변수로 변환하여 케환을 제거한다. 케환이 제거된 회로는 DAG로 구성된 후, 출력 에지의 개수가 2 이상인 노드가 존재할 경우 노드만을 복제하여 DAG을 팬 아웃 트리 트리로 재구성함으로써 회로의 전체 지연 시간이 최소화 될 수 있도록 하였다. 또한 시간제약 조건과 소자의 지연시간을 이용하여 분할 가능한 단단의 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 기술매핑 알고리즘이 기존의 CPLD 기술 매핑 툴인 TEMPLA

에 비해 수행시간은 4.64% 증가되었으나 면적에서는 DDMAP에 비해 69.03%, TEMPLA에 비해 15.58% 감소된 결과를 나타내었다.

향후 연구 과제로서 CPLD를 구성하고 있는 매크로셀의 내부가 2중 구조인 Heterogeneous한 구조에 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] 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.
- [5] 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.
- [6] 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.
- [7] J. L. Kouloheris, "Empirical Study of the Effect of Cell Granularity on FPGA Density and Performance", Ph. D. Thesis, Department of Electrical Engineering Stanford University, 1993.
- [8] 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.
- [9] J. H. Anderson and S. D. Brown, "Technolog

Mapping for Large Complex PLDs”, Proceeding of the 35th DAC, pp698-703, 1998.

[10] 김희석, 변상준 "EDIF Netlist를 이용한 PLD 설계용 툴 개발", 한국정보처리학회 논문지, 제5권, 제4호, pp1025-1032

저 자 소 개



尹 忠 模(正會員)

1954년 5월 20일생. 1985년 2월 서울 산업대 전자공학과 학사. 1990년 8월 단국대학교 전자공학과 석사. 1999년 2월 청주대학교 전자공학과 박사 수료. 1993년 3월~현재 서일대학 정보통신과 조교수. 관심분야 : CAD,

VLSI 설계, 그래픽 알고리즘



金 才 鎭(正會員)

1967년 10월 6일생. 1993년 2월 청주대학교 전자공학과 학사. 1995년 8월 청주대학교 전자공학과 석사. 1997년 9월~현재 청주대학교 전자공학과 박사과정. 관심분야 : CAD, VLSI 설계, 멀티미디어