

# TLU형 FPGA를 위한 기술 매핑 알고리즘

박 장 현<sup>†</sup> 김 보 관<sup>††</sup>

## 요 약

본 논문은 새로운 ASIC 구조로 최근에 관심을 모으고 있는 Table Look-Up형의 FPGA를 위한 기술 매핑에 대한 연구에 관한 것이다. 이를 위해 우선 다단계 논리 합성, decomposition reduction, packing 등 각 설계 과정의 기존 알고리즘을 비교 분석하였고, 각 과정에서 새로운 알고리즘을 첨가하였다. 설계 시스템의 최적화 대상인 CLB 갯수 및 네트 수의 최소화를 동시에 고려하기 위하여 사용자에게 의해서 주어지는 각 요소의 가중치 선형함수로 된 비용 함수를 제안하였다. 이 비용함수를 사용하여 벤치마크 테스트를 통한 일반적인 회로에 적용성 검증, 기존 알고리즘과의 성능 비교 및 개선에 대해 연구하였다. 논리 설계 합성기 구성 과정에서 Node-pair decomposition, merging fanin, unified reduction, 다출력 decomposition 등 4가지의 새로운 알고리즘을 추가하여 실험한 결과, CLB와 네트의 수에서 SIS-pga보다 약 10% 적은 값을 얻을 수 있었다.

## Logic Synthesis Algorithm for TLU-Type FPGA

Jang-Hyun Park<sup>†</sup> and Bo-Gwan Kim<sup>††</sup>

## ABSTRACT

This paper describes several algorithms for technology mapping of logic functions into interesting and popular FPGAs that use look-up table memories. In order to improve the technology mapping for FPGA, some existing multi-level logic synthesis, decomposition reduction and packing techniques are analyzed and compared. And then new algorithms such as node-pair decomposition, merging fanin, unified reduction and multiple output decomposition which are used for combinational logic design, are proposed. The cost function is used to minimize the number of CLBs and edges of the network. The cost is a linear combination of each weight that is given by user. Finally we compare our new algorithm with previous logic design technique[8]. In an experimental comparison our algorithm requires 10% fewer CLB and nets than SIS-pga.

## 1. 서 론

FPGA(Field Programmable Gate Array)는 Carter[1] 등에 의해 제안된 가장 새로운 ASIC 구조로, 기존의 MPGA(Mask Programmable Gate Array)와 같이 배선영역을 사이에 둔 로직 셀의 2차원적 배열이나, PLD와 같이 로직 셀의 기능과 셀들 간의 상호연결을 사용자가 임의로 결정할 수 있도록 되어있다. 이와 같이 기존의 두 종류 구조를 혼합함으로써 FPGA는 짧

은 제작 시간, 저렴한 제작 가격, 제작과 테스트 용이성 및 고집적도 등 PLD와 MPGA의 장점을 두루 갖추고 있어, 시제품 제작 및 소량생산에 매우 적합한 소자이다.

FPGA 구조에 회로를 구현하기 위한 연결은 정해진 배선 영역을 사용하여 프로그램되어야 하며, 블록 구조에 따라 TLU(Table Look-Up)형과 MB(Multiplexer-Based)형으로 분류한다. TLU형 구조에서 로직의 기본 단위를 CLB(Configurable Logic Block)[2]라고 부른다. TLU형의 기본 블록은 m개의 입력으로 어떤 함수도 구현 가능토록 되어있으며, TLU 구조가 결정되면 m은 항상 고정되어 있다. 이 구조의 전형적인

<sup>†</sup> 정 회 원 : 한국전자통신연구소 선임연구원

<sup>††</sup> 정 회 원 : 충남대학교 전자공학과 부교수

논문접수 : 1995년 4월 25일, 심사완료 : 1995년 10월 4일

에는 Xilinx(XC3000)[2] 구조로서 이 경우  $m=5$ 이다. 로직 블럭들간의 연결부는 프로그램 가능한 트랜지스터로 구성되어 있으며, 이들 로직 함수와 로직 블럭간의 상호연결은 내부 정적 메모리에 저장되어 있는 프로그램 데이터에 의해서 결정된다. 위와 같은 특성을 가진 FPGA의 기술 매핑 관점에서 중요 제약 조건은 i) 한 칩에서 제한된 CLB의 갯수 ii) 한 CLB가 가질 수 있는 최대 입력수 iii) 제한된 배선영역 iv) 입력에서 출력까지 지연시간 등이다.

MISII[3]나 GLSS[4] 등의 다단계 로직 합성 도구에서는 면적의 평가로서 게이트의 수나 불리안 회로의 노드에서 문자 수를 사용하였다. 그러나, 이것은 CLB의 수를 최소화하는 평가 방법에는 적합치 않다. 왜냐하면 아무리 복잡한 표현식을 갖는 함수일지라도 최대 입력변수의 수에 대한 조건만 만족된다면 하나의 CLB로 구현될 수 있기 때문이다. 예를 들면,  $m=5$ 이고,  $f1=abcdeg$ ,  $f2=abc+b'de+a'e'+c'd'$ 라 하자. 함수  $f1$ 과  $f2$ 는 각각 6개, 10개의 문자 수를 가지고 있다. 함수  $f1$ 는 그것의 최적 구현에서 2개의 CLB가 요구되나,  $f2$ 는 5개의 변수만 갖고 있으므로 한개의 CLB로 구현이 가능하다. 이와같이 목적함수는 함수를 구현하는 실제적 로직보다도 입력의 수에 중점을 두어야만 한다. 전체 회로에서 입력이  $m$ 보다 작은 중간 노드의 수가 이 회로에 필요한 CLB 수의 상한 갯수이다. 기본적인 문제는 회로가 주어지면 목표 FPGA 구조의 기본 블럭을 사용하여 회로를 구현하는 것이다. 주어진 회로를 구현하는 과정에서 최종 목적은 사용된 블럭수의 최소화, 지연 시간 최소화, 배선 영역의 최소화 등이다.

종래의 연구에서 FPGA 기술 매핑을 위한 여러가지 방법[5, 7, 8, 10, 11, 12, 13, 14, 15]이 연구되었지만 아직도 알고리즘 구현방법과 성능 면에서 완전히 만족할 수 없는 상황이다. 따라서 본 연구에서는 새로운 ASIC 구조로 최근에 관심을 모으고 있는 TLU형의 FPGA를 위한 더욱 더 성능이 좋은 로직설계 시스템의 개발을 목적으로 한다. 본 시스템을 개발하고자하는 이유는 아직도 FPGA 구조상 CLB 수와 배선 영역이 매우 제한적이기 때문에 제한된 이들 자원을 극

대화하는 것이 제품의 가치를 배가시키는 것이기 때문이다. FPGA 로직 설계를 위한 본 연구에서는 기본적으로 2단계 접근 방법을 따르기로 한다. 물론 각 과정에서의 구체적인 작업 내용은 FPGA의 특성이 잘 고려되어야 할 것이다. 특히 TLU 구조의 FPGA에서는 두 과정이 서로 밀접한 상관 관계가 있다. 본 연구에서 구현하고자 하는 시스템의 입출력 및 최적화 조건은 다음과 같다.

- 1) 입력 : 각 노드가 곱의 합 표현식으로 주어지는 불리안 회로  $N$ 과 CLB 입력 변수  $m$
- 2) 출력 : 각 노드가  $m$ 개 이하의 변수를 사용한 곱의 합 표현식인 다단계 회로  $N'$
- 3) 최적화 내용 : CLB의 갯수, 배선 복잡도

입력이 다단계로 주어졌을 때에는 이미 전체 로직구조가 잘 설명된 것으로 가정하여 바로 기술 매핑 단계를 수행한다. 최적화 내용 중에서 면적은 CLB 갯수로, 배선 복잡도는 네트 수로 근사화한다. 최적화 대상인 CLB의 갯수, 네트 수 및 지연시간을 동시에 고려하기 위하여 비용함수는 사용자에게 의해서 주어지는 가중치  $K1$ ,  $K2$ ,  $K3$ 에 의한 선형합을 제안한다.

$$\text{비용함수} = K1(\text{면적}) + K2(\text{네트수}) + K3(\text{지연시간})$$

이 비용함수를 이용하여 주어진 회로의 비용을 줄이는 최적의 TLU형 FPGA 기술 매핑을 구한다. 이런 기술 매핑 시스템 개발을 위한 논문 구성 순서는 지금까지 설명한 서론에 이어 2장에서 기존 FPGA 기술 매핑 알고리즘을 분석하고, 3장에서는 4가지 새로운 기술을 이용한 FPGA 기술 매핑 알고리즘에 대해서 설명한다. 4장에서는 각 알고리즘에 대한 실험 결과를 보여주며, 마지막 5장에서는 결론을 내린다.

## 2. 기존 알고리즘 분석

FPGA를 위한 로직설계 연구는 최근 국내외에서 많은 연구 결과가 발표되는 새로운 분야이다. 일반적으로 FPGA를 위한 시스템 개발은 decomposition, reduction, packing 등 3단계로 분류되어 있다. 연구의 각 분야는 위의 세단계를

적절히 조합하여 다루는 것이다. 그 중 한분야는 이미 다단계로 최적화된 불리안 네트워크를 입력으로 받아서 이를 목표로 하는 FPGA 구조로 변환하는 FPGA 기술 매핑에 관한 것으로, 로직 셀의 기능이 고정되어 있지 않고 사용자가 임의로 결정할 수 있는 특성을 최대한 활용한다. 또 다른 분야는 2단계에서 다단계로의 전개시 목표로 하는 TLU FPGA의 특성을 고려하는 연구로서 각 노드에 포함되는 변수의 갯수를 최소화하는 방법과 기존의 다단계 합성 방법에서 비용함수만 변화시킨 방법이 있다.

MIS-pga(old)[5]: MIS-pga(old)에서는 decomposition 과정에서 non-feasible 노드를 feasible 노드로 변환하는 여러가지 방법중 Roth-Karp decomposition 방법[6]과 분할을 사용하였다. 그 다음 적당한 collapsing 방법을 통하여 주어진 feasible 로직 회로에서 어떤 노드를 제거한 회로가 feasible하면서 노드의 수를 최소화 하는 covering 문제로 변환한다. 이렇게 구한 결과를 FPGA의 특수한 구조를 이용하여 합병 가능한 함수의 disjoint 짝의 가장 큰 집합을 구한다. 이 프로그램 수행은 분할, simplify, cover, merge의 순서를 따랐으며 실험 결과 cover가 효과적이지만 비용이 많이 드는 함수였다. 작은 회로에 대해서는 정확한 알고리즘이 사용되었지만, 크기가 큰 회로에서는 휴리스틱한 방법을 사용하였다.

MIS-pga(new)[7]: MIS-pga(new)는 MIS-pga에 decomposition 방법으로 bin packing과 cofactoring 방법을 첨가하였으며, 2 출력 CLB를 위해 cover-merge를 같이 수행할 수도 있다.

SIS-pga[8]: SIS-pga는 순차회로 로직 설계 시스템으로서 조합회로 로직 설계 시스템인 MIS-pga의 모든 기능을 포함하면서, 순차회로를 위한 로직 설계 기능도 포함하고 있다.

Chortle-crf[10]: Chortle[9]은 주어진 회로를 트리 구조로 변환한 후 각 트리마다 최적의 매핑 해를 구했다. 트리 구조의 제약 때문에 재결합하는 변수가 있을 경우 효과적인 처리는 불가능했다. Chortle-crf는 bin packing 접근 방식과 재결합 경로의 특성을 응용했으며, 다출력인 경우 필요하면 게이트의 복제도 허용했다.

TechMap[11]: 주 입력에서 주 출력으로 매핑을 해가면서, 기 매핑된 CLB간의 compatibility형과 clique 분할 및 Shannon cofactoring에 기초한 TLU형 FPGA 기술 매핑(mapper)로서, 면적과 지연 시간을 최소화한다. 이 알고리즘의 주안점은 한 노드를 TLU로 매핑할 때, 그 노드에 대한 입력을 줄임으로써 그 노드를 구현하기 위해 새로이 생성하는 TLU의 수를 최소화하는 것이다. 이 과정에는 기 매핑된 입력 TLU들 중 합병가능한 두 TLU를 하나의 TLU로 대체하는 변환과 TLU의 감소없이 단순히 입력만 줄이는 변환이 사용 된다. 가능한 변환이 많을 때 clique 분할 알고리즘을 사용하여 최적의 변환 집합을 선택한다.

Asyl[12]: 기술 매핑은 로직 최적화 과정에 의해서 생성된 lexico graphical factorization 네트워크를 사용하여 결정되는 주 입력변수의 순서에 의존한다. 이 시스템의 기술 매핑 과정은 2단계로 이루어져 있으며 첫 단계는 불리안 회로를 4 혹은 5 입력 회로로 만들며, 두번째 단계에서는 greedy 알고리즘을 사용하여 이 함수를 CLB에 매핑시킨다.

Hydra[13]: 2-output CLB 사용에 중점을 두고 개발한 시스템으로 공동 입력 그래프를 데이터 구조로 한 functional decomposition을 기본으로하여 합병과 매핑을 동시에 수행한다.

Xmap[14]: CLB의 수를 최소화하기 위한 두 단계와 CLB로 매핑하는 한 단계로 구성되어 있다. 첫번째 단계는 주어진 회로를 if-then-else DAG로 바꾸며 입력이 3 이하이면 모든 노드가 한 CLB로 구성 가능하다. 두번째 단계는 한 출력 TLU로 구현가능토록 각 노드를 decomposition 한다. 세번째 단계는 2 출력 TLU를 포함한 최소한 CLB 갯수를 구한다.

VISMAP[15]: 주어진 입력 네트워크의 모든 노드가 이미 기본 블록의 입력수 이하인 feasible 네트워크에서의 노드 reduction과 packing을 위한 covering 문제에 중점을 둔 기술 매핑이다. 그래프의 각 네트 마다 visible 혹은 invisible로서 표시하여, visible 네트는 2 TLU를 연결하고, invisible 네트는 한 TLU 내부에 연결된다. 주어진 회로를 m개의 네트를 가진 작은 그래프로 분리

한 후 exhaustive 조사 방법을 사용하여 각 네트에 최적의 레이블 값을 준다. VISMAR는 네트의 visible를 기초로 하여 CLB와 네트의 숫자를 최소화 하였다.

### 3. FPGA 매핑 알고리즘

TLU형 FPGA의 로직 설계는 입력으로 주어지는 2단계 또는 다단계 회로의 로직 구조를 최적화하는 로직 최적화와, 이것의 결과를 대상 FPGA의 구조로 변환하는 기술 매핑의 순차적인 두 과정으로 이루어진다. 기술 매핑에서는 다시 1) infeasible 네트워크를 각 노드가 하나의 블럭으로 1-대-1 대응될 수 있는 feasible 네트워크로 변환하는 decomposition, 2) feasible 네트워크에서 노드 수, 네트 수, 레벨 수 등을 최소화하는 reduction, 3) 결합 가능한 두 함수를 하나의 CLB로 구현하는 packing의 세 과정이 순차적, 반복적, 또는 복합적으로 행해진다. 각 과정마다 여러가지의 네트워크 변환들이 사용된다. 가장 대표적인 TLU형 FPGA 로직 합성 시스템인 SIS-pga[8]은 전반적인 시스템의 구성을 script-base이며 표준 script에 의한 batch 모드뿐 아니라 interactive 모드로도 동작시킬 수 있다. 이렇게 하므로써 세부 알고리즘 개발에 의한 시스템의 성능개선, 각종 휴리스틱의 장단점 검토 및 특별한 경우를 처리하기 위한 모듈의 추가가 용이하다. 본 연구에서는 SIS-pga[8]에 node-pair decomposition, merge-fanin, unified reduction, 다출력 Roth-Karp decomposition 등의 새로운 변환 알고리즘들을 추가하여 로직 합성기의 성능을 개선하고자 한다.

#### 3.1 Node-pair Decomposition

Node-pair decomposition은 다단계 로직구조 합성과정에서 사용된 kernel 추출 등의 algebraic 방식이 발견하지 못하였을 공통 함수를 불리안적인 방법을 사용하여 추출하는 것을 목적으로 한다. (그림 1)은 이 변환 알고리즘의 요약이다. 먼저 주어진 회로로부터 공통의 입력신호가 2개 이상인 infeasible 노드 쌍들(S)을 구한 후, 공통 입력 신호가 가장 많은 쌍부터 순차적으로

SDD(simple disjoint decomposition)의 존재여부를 조사한다. SDD가 존재하면 관련된 두 infeasible 노드들을 분해된 표현식으로 대체하며, 아울러 두 노드를 포함하는 후보들을 이후의 고려 대상에서 제외한다. 이와 같은 과정을 더 이상 고려 대상이 없을 때까지 반복한다.

```
Node-pair dec(network, support, limit, mode, A bound){
  S={(n1, n2, C) | n1 and n2 are infeasible nodes,
    C= | fanins(n1) ∩ fanins(n2) |, where C>=2};
  Repeat
    Peak-out s∈S with the largest C;
    If(|C| > limit) then continue to the next s;
    Try decomp. of n1 & n2 by some subset c∈C;
    If (possible) {
      Change the network;
      Delete s∈S if n1 or n2 is a member;
    };
  Until(S becomes empty);
}
```

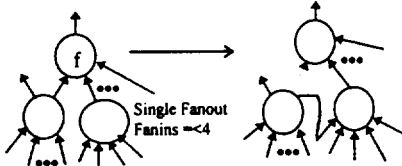
(그림 1) Node-Pair Decomposition 알고리즘  
(Fig. 1) Procedure of Node-Pair-Decomposition

입력신호수가 지원변수보다 크면 infeasible 노드이다. SDD의 조사는 최악의 경우 공통 입력 신호의 모든 부분 집합에 대해 행하여지며, 따라서 상당한 계산시간을 요하는 연산이다. Limit는 이러한 경우를 제어하기 위한 파라미터로써, 공통 입력 신호의 수가 limit보다 크면 그 노드 쌍에 대해서는 SDD를 조사하지 않는다. Mode는 decomposition의 한계 집합을 찾는 방법을 설정하기 위한 것으로써 mode=0(최적의 해 선택), mode=1(현재 이득과 같으면 제외), mode=2(첫번째 SDD 선택)이다. Mode가 2일 때의 조사 순서는 해당 한계 집합을 전체 변수의 부분 집합으로 표현했을 때의 2진 표현 순서에 의한다. A-bound는 해당 한계 집합으로 decomposition했을 때의 알파 함수에 대한 제한으로 한계 집합과 함께 네트워크를 구성할 CLB의 수를 결정하게 된다.

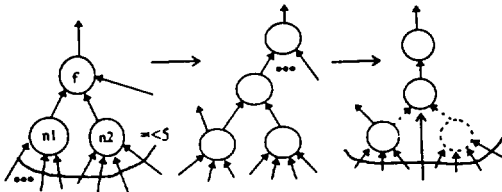
#### 3.2 Merge Fanins

새로운 노드의 생성없이 어떤 노드의 입력을 줄일 수 있다면, 그 노드의 infeasibility를 감소시킬 뿐만 아니라, 그 노드를 사용한 합병 가능

성을 증가시킬 수 있다. MIS-pga의 move-fanin (그림 2)은 변환된 상태 자체가 이전 회로보다 네트의 수가 하나 줄었을 뿐만 아니라 노드 f의 입력이 하나 줄었기 때문에 추후에 노드 f를 그의 출력에 병합하여 그 노드를 제거할 수 있는 가능성을 증가시킨 점이다.



(그림 2) Move-Fanin 변환  
(Fig. 2) Move-Fanin Transformation



(그림 3) Merge-Fanin 변환  
(Fig. 3) Merge-Fanin Transformation

그러나 이 move-fanin 변환은 원래의 회로보다 레벨을 하나 증가시키므로, 지연 시간을 고려한 경우에는 move-fanin 노드가 임계 경로에 있어서는 안 된다. 제안된 merge-fannins(그림 3) 변환은 적용이 되는 경우 레벨의 증가없이 입력을 줄일 수 있다. (그림 4)에 merge-fanin 알고리즘을 보여주고 있다.

```

Merge fanin {
  for each subset N ⊆ fanins(f) {
    if (∪_{n ∈ N} fanins(n) > 5) continue;
    N' = {n ∈ N | internal, single-fanout};
    Decompose f by bound-set N;
    f = g(α(N), ..., α_{max}(N)) with α_{max} = <min(N', N-1);
    if (possible) {
      add α to the network;
      replace f by g;
      collapse all n ∈ N to α;
    }
  }
}
    
```

(그림 4) Merge-Fanins 알고리즘  
(Fig. 4) Procedure of Merge-Fanins

위 (그림 4)의 알고리즘에서 보듯이 두 노드의 입력 변수가 5보다 작고 이들 노드가 하나의 출력을 갖고 있을 때 이 노드들을 합병하여 하나의 노드로 만든다. 이렇게 합병된 노드(그림 3)가 함수 f의 입력으로 사용되면 네트의 수가 하나 줄었고, 함수 f의 변수도 하나 줄었기 때문에 f의 합병 가능성을 높혀준다.

### 3.3 Unified Reduction

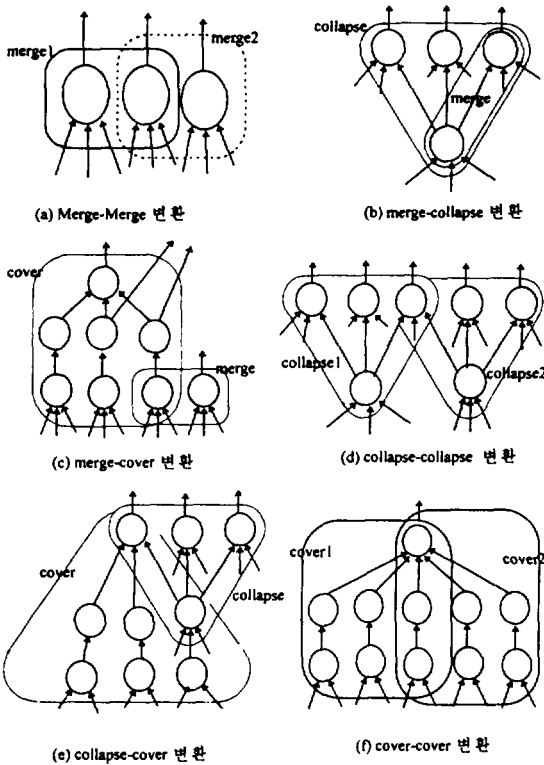
Feasible 네트워크에서 CLB와 네트의 갯수를 줄이는 대표적인 변환으로는 collapse, packing, cover가 있다. Collapse는 cover의 간단한 형태로써 cover는 전체적인 정보를 사용하고 collapse는 오직 지역적인 정보만을 사용한다. MIS-pga에서는 이들 세 변환이 개별적인 명령어에 의해서 독립적으로 행해진다. 각 변환은 네트워크를 조사하면서 순차적으로 행해지거나, 그 변환이 적용가능한 모든 노드 집합들을 구한 후 이로부터 그 변환을 적용할 최적의 부분 집합을 구한다. 그러나 세 변환 중 어떤 변환이 더 바람직한지의 여부와 이에 따른 적용 순서는 결정하기 매우 어려운 문제이다. Unified reduction은 이들 변환을 종합적으로 고려하고, 각 적용가능한 변환들이 비용에 미치는 영향과 이들 사이의 incompatibility를 고려하여 최적의 변환 집합을 구하고자 한다. (그림 5)에 그 과정을 보여주고 있다.

```

Procedure Unified_reduction{
  TL = {t | possible transformation with positive gain G(t)};
  Get incompatible transformation I(t) for each t ∈ TL;
  G'(t) = G(t) - ∑_{s ∈ TL} G(s);
  Repeat
    pick t ∈ TL with the best G'(t) and apply t;
    Mark out s ∈ I(t) from TL;
    Update incompatibility I(t)'s;
    Update G'(t); /* G(t) not updated */
  Until(TL is all marked)
}
Uni_red_app{
  repeat (reduction) until (network cost goes down);
  apply cost-preserving transformation;
  repeat (reduction) until (network cost goes down);
}
    
```

(그림 5) Unified\_Reduction 알고리즘  
(Fig. 5) Procedure of Unified\_Reduction

먼저 merge, collapse, cover의 세 변환에 대하여 적용이 가능한 경우와 각각의 경우의 비용 효과를 구한다. 이들간의 우선순위를 결정하기 위하여 각 변환간의 incompatibility를 구하여, 자체 변환의 이득에서 incompatible한 변환들의 이득을 제외한 것을 새로운 이득으로 정의하여 이 새로운 이득이 제일 큰 것을 선택하여 적용한다. 일단 하나의 변환이 선택되어 적용되면 이것과 incompatible한 변환들은 고려대상에서 제외한다. 이와 같은 greedy 알고리즘의 최적 적용을 위해서는 매 변환 이후에 새로운 변환 집합과 이득을 계산하는 것이 바람직하지만, 계산시간을 고려할 때 거의 불가능한 방법이다. 따라서, 본 연구에서 채택한 방법은 변환 집합과 각 변환 자체의 이득 계산은 그 과정의 시작 단계에서 구한 것을 그대로 사용하고, 변환적용에 의한 incompatibility의 변경과 이에 따른 상대이득의 변화만을 고려한다.



(그림 6) Incompatible한 두 변환들  
( Fig. 6) Two Incompatible Transformations

하나의 변환을 적용하면 이에 따라 이전에 불가능했던 변환이 새로이 적용 가능해질 수가 있는데, 매 변환 적용 이후에 새로운 변환을 추가하는 것은 너무 계산시간이 많이 걸리기 때문에 행하지 않는다. 대신에 이러한 단점을 보완하기 위하여, 일단 unified reduction을 수행한 이후에 전체회로의 비용이 유지 또는 증가되지 않는 범위 내에서 네트워크를 변화시킨 후 unified reduction을 다시 한번 수행한다. 상술한 알고리즘의 핵심은 적용가능한 변환들 간의 동시적용 가능성을 고려한 선택으로써, incompatibility의 정의 및 계산은 이 알고리즘의 효과 및 효율성을 결정하는 요소이다. 두 변환을 동시에 적용 불가능할 때 이 두 변환은 incompatible하다고 하며, 두 변환의 종류에 따른 incompatibility 판정은 계산의 복잡도를 고려하여 다음과 같은 경우에 한한다.

- (1) merge-merge 변환 : 두 merge 대상 노드 집합들간에 공통된 노드가 있을 때((그림 6 (a)).
- (2) merge-collapse 변환 : (a) collapse 변환에 의해 제거될 노드를 포함하는 합병 변환 또는 합병 대상 두 노드 중 하나 또는 두 노드 모두가 collapse에 의해 변경될 경우((그림 6 (b)).
- (3) merge-cover 변환 : cover와 합병 대상 노드집합간에 공통된 노드가 있을 때((그림 6 (c)).
- (4) collapse-collapse 변환 : 두 병합 대상 노드 집합들간에 공통된 노드가 있을 때에는 하나의 병합 변환 이후에도 다른 변환이 여전히 가능한지의 여부를 조사하여 판정((그림 6 (d)).
- (5) collapse-cover 변환 : 병합과 cover의 노드간에 공통된 노드가 있을 때((그림 6 (e)).
- (6) cover-cover 변환 : 동일한 노드를 sink 노드로 하는 두 cover 변환((그림 6 (f)).

### 3.4 다출력 Roth-Karp Decomposition

다단계 로직구조 합성의 전처리과정과 TLU로

의 분해 변환에서 사용한 algebraic 방법의 단점 보완을 주목적으로 SIS-pga[8]에서는 두 번의 불리언 decomposition을 시도한다. 즉 지금까지의 모든 로직합성 결과를 무시하고 네트워크를 2 단계로 완전히 병합한 후 Roth-Karp 방법과 cofactoring 방법의 두 가지 불리언 decomposition만으로 각각 설계하여 algebraic 방법에 의한 결과를 포함한 세 가지 결과 중에서 제일 좋은 결과를 선택한다. 그러나 SIS에서 채택한 방법은 각각의 노드에 대해 개별적으로 decomposition을 행하기 때문에, 공통된 중간 노드의 생성을 보장하지 못한다. 이러한 단점을 보완하기 위하여 본 연구에서는 제안한 다출력 Roth-Karp decomposition은 Roth-Karp 방법을 다출력 함수에 적용할 수 있도록 다음과 같이 확장한 것이다.

```
Mul out decomp{
  N={n an infeasible node};
  P={{(N', F') N' ⊆ N, F' = fanins common to N'}};
  repeat
    pick a maximal p=(N', F') (P with max |N'|);
    for (each subset F ⊆ F', F=|F'|){
      decompose N' by F with a_max < F
      N' = (a_1(F), ..., a_max(F));
      if (possible) change the network;
    }
  until (a decomposition found);
}
Repeat this when a decomposition is found.
if (still not feasible) do split network( ).
```

(그림 7) 다출력 Decomposition 알고리즘  
(Fig. 7) Procedure of Multiple Output Decomposition

(그림 7)은 이 알고리즘 과정을 설명한 것이다. 먼저 infeasible한 노드 중에서 공통의 입력 신호를 갖는 함수들을 찾는다. 관련 함수가 많고 공통변수의 수가 많은 집합부터 decomposition의 존재여부를 조사하며, 만일 존재하면 이 변환에 의한 이득을 계산하여 임시 저장하고, 추후에 발견되는 또 다른 decomposition과 비교하여 최적의 것을 찾을 수 있도록 한다. 일단 하나의 decomposition이 선택되어 수행됨으로써 네트워크가 변화되면 infeasibility 등 모든 상황이 변화되므로 다시 공통변수를 갖는 infeasible 노드 집합을 조사하여야 한다. 이러한 과정을 바람직한 Roth-Karp decomposition이 발견되지 않을 때

까지 반복 수행하고, 마지막으로 네트워크가 여전히 infeasible할 때에서 적절한 방법을 통해 feasible 네트워크로 변환한다.

#### 4. 실험 결과

본 연구에서는 UC-Berkeley의 SIS-pga[8] 표준 Script에 제3장에서 설명한 4가지 변환 알고리즘, 즉 Node-Pair Decomposition, Merging Fanin, Unified Reduction, 다출력 Decomposition의 새로운 변환을 추가하여 CLB 갯수, 네트워크의 수, 지연 시간을 동시에 최적화 하도록 하였다. CLB의 수, 네트워크의 수 및 지연 시간에 대한 비용 함수의 가중치는 K1, K2, K3를 각각 1로 하여 동등한 비중을 갖도록 하였다. 실험에 사용된 회로는 31개 회로로써, 기존의 알고리즘과의 비교를 위하여 각 문헌에서 사용한 회로 중 표준 테스트 회로에 있는 것을 모두 취합한 것이다. 각 알고리즘별 결과는 다음과 같으며, SIS-pga와 CLB 갯수 또는 네트워크의 수에서 차이를 보인 경우만을 나열하였다. 결과 테이블은 CLB 수, 네트워크 수 및 레벨 수를 표시하였다.

##### 4.1 Node-Pair Decomposition 결과

(표 1)에서 첫번째 행은 테스트 회로와 입력력을 나타내며, 두번째 행은 SIS-pga의 표준 script로 수행한 결과이며, 세번째 행은 SIS-pga의 표준 script에 node-pair decomposition 알고리즘을 첨가하여 수행한 결과이다. 수행 방법은

(표 1) Node-Pair Decomposition의 결과  
(Table 1) Results of Node-Pair

Circuits (IN OUT)	SIS Only			Node-Pair 추가		
	CLB	NET	LEV	CLB	NET	LEV
C880(60 26)	71	340	10	69	332	9
apex2(39 3)	73	354	9	74	362	9
alu4(14 8)	88	431	15	94	462	21
duke2(22 29)	97	478	7	98	484	7
sao(10 4)	32	153	7	32	156	9
misex2(25 18)	27	130	3	27	126	4
clip(9 25)	20	96	6	19	87	5
inc(7 9)	24	117	5	29	139	5
Total	432	2,095	62	442	2,148	69

Decomposition

표준 script 12단계 중 3단계와 4단계 사이에 node-pair 알고리즘을 넣고 13단계를 수행한 것이다. 전체 알고리즘 수행 초기에 본 알고리즘을 추가하여 전처리기로 사용 가능성을 찾고자 하였다. <표 1>에서 보듯이 node-pair decomposition한 결과는 CLB의 갯수는 432에서 442로, 네트 수는 2099에서 2148로, 레벨 수는 62에서 69로 증가하였으며 비용함수 중 어느 부분도 향상된 것은 없었다.

4.2 Merging Fanin 결과

<표 2>에서 첫번째 횡은 테스트 회로와 입출력을 나타내며, 두번째 횡은 SIS-pga의 표준 script로 수행한 결과이며, 세번째 횡은 SIS-pga의 표준 script에 merge-fanin 알고리즘을 첨가하여 수행한 결과이다. 수행 방법은 표준 script 12단계 중 3과 4사이에 merge-fanin을 넣고 13단계를 수행한 것이다. 전체 알고리즘 수행 초기에 본 알고리즘을 추가하여 가능하면 merge-fanin의 갯수를 많아 찾고자 하였다. <표 2>에서 보듯이 merge fanin한 결과는 CLB의 갯수, 네트 수, 레벨 수에서 약간의 비용함수 향상을 가져왔다.

4.3 Unified Reduction 결과

<표 3>에서 첫번째 횡은 테스트 회로와 입출

력을 나타내며, 두번째 횡은 reduction을 하기의 결과이며, 세번째 횡은 SIS-pga의 표준 script로 수행한 결과이고, 네번째 횡은 SIS-pga의 표준 script 12단계중 10-12단계의 collapse, cover, merge 대신에 제안된 알고리즘 unified reduction을 첨가하여 수행한 결과이다.

4가지 추가된 변환 알고리즘중 가장 중점적으로 연구하였으며, <표 3>의 결과에서 나타나듯이 CLB수와 네트수의 최소화에 상당히 효과적이었다. 그러나 레벨수는 약간 증가하였다.

<표 3> Unified Reduction의 결과  
(Table 3) Results of Unified Reduction

Circuits (IN OUT)	Reduction 전			SIS			Uni-Reduction		
	CLB	NET	LEV	CLB	NET	LEV	CLB	NET	LEV
count(35 16)	31	121	10	28	124	5	27	114	10
apex7(49 37)	60	234	6	46	219	6	41	193	6
e64(65 65)	80	300	17	56	252	17	56	250	17
C880(60 26)	93	396	12	71	340	10	70	337	9
apex2(39 3)	89	352	9	73	354	9	69	334	9
alu2(10 6)	108	440	18	88	431	15	81	389	16
duke2(22 29)	117	442	10	97	478	7	92	398	10
C499(41 32)	70	264	7	62	288	7	51	245	7
rot(135 107)	186	687	16	142	687	15	128	619	16
apex6(135 99)	211	831	10	175	861	10	162	760	10
alu4(14 8)	199	817	22	171	835	21	151	736	22
misex2(25 18)	31	126	4	27	130	3	25	119	4
bw(5 28)	42	177	13	33	162	5	25	119	5
inc(7 9)	32	133	7	24	117	5	23	110	5
Total	1,349	5,320	35	1,093	5,278	135	981	4,723	146

<표 2> Merge-Fanins의 결과  
(Table 2) Results of Merge-Fanins

Circuits (IN OUT)	SIS Only			Merge-Fanins 추가		
	CLB	NET	LEV	CLB	NET	LEV
count(35 16)	28	124	5	28	123	5
apex(49 37)	46	219	6	45	212	6
apex2(39 3)	73	354	9	74	361	9
duke2(22 29)	97	478	7	97	475	6
rot(135 107)	142	687	15	142	694	12
apex6(135 99)	175	861	10	177	876	10
alu4(14 8)	171	835	21	159	774	22
sao2(10 4)	32	153	7	33	159	8
misex2(25 18)	27	130	3	26	124	3
inc(7 9)	24	117	5	23	114	5
clip(9 25)	20	96	6	23	113	5
b9(41 21)	27	129	4	27	130	4
b12(15 9)	16	79	3	16	76	3
Total	878	4,262	101	870	4,231	98

<표 4> 다출력 Roth-Karp 분해 결과  
(Table 4) Results of Multiple Output Roth-Karp Decomposition

Circuits (IN OUT)	Cofactoring			Both-Karp			Multiple-Output		
	CLB	NET	LEV	CLB	NET	LEV	CLB	NET	LEV
z4ml(7 4)	13	63	3	8	38	2	10	46	3
misex1(8 7)	18	85	3	14	63	2	10	45	3
5xpl(7 10)	21	101	3	17	80	2	11	51	3
9symml(9 1)	24	118	5	7	35	3	6	20	3
rd84(8 4)	37	184	4	12	58	3	7	28	3
rd73(7 3)	17	83	3	7	34	2	5	21	2
f51m(8 8)	23	110	4	20	98	4	10	45	4
clip(9 5)	96	449	5	83	378	17	78	374	16
inc(7 9)	23	112	3	27	128	2	20	99	3
con1(7 2)	4	18	2	4	16	2	3	13	2
Total	276	1,323	35	199	928	39	160	742	42



4.4 다출력 Decomposition 결과

〈표 4〉에서 첫번째 횡은 테스트 회로와 입출력을 나타내며, 두번째 횡은 cofactoring decomposition 결과이며, 세번째 횡은 Roth-Karp decomposition 결과이고, 네번째 횡은 다출력 Roth-Karp decomposition 결과이다. 세 경우 모두 SIS-pga의 표준 script 12단계중 처음 3단계만 수행하고 〈표 4〉에 나타난 3가지 방법을 수행한 것이다. 〈표 4〉에서 나타나듯이 다출력 decomposition도 CLB와 네트 수에서 비용함수 향상을 보였으나, 레벨수는 약간 증가하였다.

4.5 지연시간을 고려한 결과

〈표 5〉에서 첫번째 횡은 다출력을 가진 테스트 회로와 입출력을 나타내며, 두번째 횡은 SIS-pga의 표준 script로 수행한 결과이고, 세번째 횡은 SIS-pga의 표준 script 12단계중 10-12단계의 collapse, cover, merge 대신에 제안된 알고리즘 unified reduction을 첨가하여 수행한 결과이다. 회로의 성능 향상을 위해 회로의 레벨수를 줄이는데 중점을 두고자, 비용 함수의 가중

치를 CLB 갯수  $K1=1$ , 네트 수  $K2=1$ , 레벨 수  $K3=3$ 으로 하여 새롭게 추가된 unified reduction 알고리즘을 수행하였다. 그러나 〈표 5〉에서 보듯이 레벨 수에는 아직 큰 효과가 없음을 알 수 있다.

5. 결 론

본 연구에서는 새로운 ASIC 구조로 최근에 관심을 모으고 있는 TLU 형의 FPGA를 위한 로직설계 합성기 개발에 대해서 설명하였다. 로직 설계 합성기를 개발하면서 최적화 대상으로는 면적, 배선 복잡도, 지연시간을 동시에 고려하기 위한 비용함수를 만들었으며, 사용자가 필요에 따라 비용함수의 각 요소별로 비중을 다르게 선택하여 사용할 수 있게 하였다. 이런 비용함수를 가진 로직 설계 합성기 구성 과정에서 Node-pair decomposition, merging fanin, unified reduction, 다출력 decomposition 등 4가지의 새로운 알고리즘을 추가하여 CLB 갯수, 네트의 수를 최적화하도록 하였으며, 기존 알고리즘과 성능을 비교 하였다. 성능 비교 결과 다출력 decomposition과 unified reduction이 매우 효과적이며, 동시에 이들을 적용하면 4장의 실험결과에서 보듯이 CLB 갯수와 네트의 수는 SIS-pga 보다 약 10%의 비용 감소를 이룰 수 있다. 아울러 앞으로 계속되어야 할 연구과제는 다음과 같다.

(1) 지연 시간을 고려한 변환 : 회로의 일부 변경이 CLB 및 네트의 수, 즉 면적에 미치는 영향은 상대적으로 국지적이고 비교적 쉽게 정량적 판단이 가능하나, 임계경로의 지연시간에 미치는 영향은 간단치 않다. 그렇지만, 고성능을 요구하는 최근의 설계요구조건을 고려할 때 지연시간을 고려한 설계기법의 개발이 필요하다.

(2) 본 연구에서 개발한 4가지 알고리즘의 적절한 조합 : 각각의 알고리즘에 의한 개별적인 영향만을 본 연구에서는 조사하였으나, 이들을 적절히 종합하여 최적의 알고리즘 조합에 대해서 연구가 필요하다.

(3) 비용함수의 각 요소가 설계에 미치는 영향 : 서로 다른 비중을 할당하여 설계자의 여러 가지 요구조건에 알맞은 회로를 생성하는지에 대

〈표 5〉 지연시간을 고려한 결과  
(Table 5) Results of Peformance

Circuits (IN OUT)	SIS Only			k1=1, k2=2, k3=3		
	CLB	NET	LEV	CLB	NET	LEV
count(35 16)	28	124	5	28	123	5
apex(49 37)	46	219	6	45	212	6
apex2(39 3)	73	354	9	74	361	9
duke2(22 29)	97	478	7	97	475	6
rot(135 107)	142	687	15	142	694	12
apex6(135 99)	175	861	10	177	876	10
alu4(14 8)	171	835	21	162	778	22
sao2(10 4)	32	153	7	34	163	7
misex2(25 18)	27	130	3	26	124	3
inc(7 9)	24	117	5	23	114	5
clip(9 25)	20	96	6	23	113	5
b9(41 21)	27	129	4	27	130	4
b12(15 9)	16	79	3	16	76	3
Total	878	4,262	101	874	4,239	96

한 추가적인 실험이 필요하다.

(4) 순차 회로에 대한 알고리즘 개발 : 지금까지의 대부분 연구가 조합 회로 구현을 대상으로 하였으나, 플립플롭을 포함하는 순차 로직회로에 대한 설계 알고리즘도 연구가 필요하다.

**참 고 문 헌**

[ 1 ] W. Carter et. al., "A User Programmable Reconfigurable Gate Array," Proc. 1986 CICC, pp.233-235, May 1986.

[ 2 ] The Programmable Gate Array Data Book, Xilinx Co., 1994.

[ 3 ] R. K. Brayton, et al, "MIS : A Multiple-Level Logic Optimization System," in IEEE Trans. CAD, pp.1063-1081, 1987.

[ 4 ] B. G. Kim and D. L. Dietmeyer, "Multi-level Logic Synthesis with Extended Arrays," IEEE Trans. on CAD, Vol.11, No.2, February 1992.

[ 5 ] R. Murgai, et al, "Logic Synthesis for Programmable Gate Arrays," 27th DAC, pp.620-625, 1990.

[ 6 ] J. P. Roth and R. M. Karp, "minimization Over Boolean Graphs," IBM Journal, pp.227-238, April 1962.

[ 7 ] R. Murgai, et al, "Improved Logic Synthesis Algorithms for Table Look Up Architectures," ICCAD, pp.564-575, 1991.

[ 8 ] R. K. Brayton, A. S. Vincentelli, et al, "SIS : A System for Sequential Circuit Synthesis," Technical Memo No. UCB/ERL M92/41, 1992.

[ 9 ] R. J. Francis, J. Rose, and K. Chung, "Chortle : A Technology Mapping Program for Lookup Table-Based Field Programmable Gate Arrays," 27th DAC, pp.613-619, 1990.

[10] R. J. Francis, J. Rose and Z. Vranesic, "Chortle-crf : Fast Technology Mapping for Lookup Table-based FPGAs," 28th DAC, pp.227-233, 1991.

[11] P. Sawkar, D. Thomas, "Area and Delay Mapping for Table-Look-Up Based Field Programmable Gate Arrays," 29th DAC, pp.368-373, 1992.

[12] P. Abouzeid, K Sakoutan, G. Saucier, and F. Poirot, "Multi-Level Synthesis Minimizing the Routing Factor," 27th DAC, pp. 365-368, 1990.

[13] D. Filo, J.C.Yang, F.Mailhot, G. De Micheli, "Technology Mapping for a Two-Output RAM-based Field Programmable Gate Array," Proc. of EDAC, pp. 534-538, 1991.

[14] K. Karplus, "Xamp : A Technology Mapper for Table-Lookup Field Programmable Gate Arrays," 28th DAC, pp.187-243, 1991.

[15] N.S. Woo, "A Heuristic Method for FPGA Technology Mapping Based on the Edge Visibility," 28th DAC, pp.248-251, 1991.



**박 장 현**

1983년 서강대학교 전자공학과 (공학사)  
 1985년 AIT 컴퓨터공학과(공학석사)  
 1994년 충남대학교 전자공학과 박사과정 수료  
 1985년~현재 한국전자통신연구

소 선임연구원  
 관심 분야 : 설계자동화, 개인통신(PCS)



**김 보 관**

1976년 서울대학교 전자공학과 (공학사)  
 1978년 국과학기술원 전기 및 전자공학과(공학석사)  
 1989년 Univ. of Wisconsin-Madison 전기 및 컴퓨터공학과(공학박사)

1980년~91년 금오공대 전자공학과 조교수  
 1991년~현재 충남대학교 전자공학과 부교수  
 관심 분야 : Logic Synthesis, Hardware/Software Codesign