

TLU형 FPGA를 위한 순차회로 기술 매핑 알고리즘

박 장 현[†] · 김 보 관^{††}

요 약

본 논문에서는 새로운 ASIC 구조로 최근에 관심을 모으고 있는 TLU형 FPGA를 위한 순차회로 기술 매핑에 관한 것이다. 본 고에서 제안하는 TLU형 FPGA를 위한 순차회로 기술 매핑방법은 먼저 6개 또는 7개의 입력을 가지는 조합 및 순차 노드에 대해서 전처리를 사용하여 한 출력 CLB에 매핑하고, 매핑안된 나머지 중 순차회로 합병 조건에 만족하는 6개 혹은 7개 입력 변수를 갖는 노드부터 CLB에 매핑시킨다. 마지막으로 앞단계에서 매핑안된 5개 이하의 입력을 가지는 노드들을 조합회로 합성기술을 사용하여 매핑한다. 본 고에서 제안한 순차회로 기술 매핑 방법이 간단하면서도 만족스런 수행시간과 결과를 얻었다. 여러개의 벤치마크 회로를 sis-pga (map_together 및 map_separate) 순차회로 합성 시스템과 비교하였으며, 실험결과는 본 시스템이 sis-pga 보다 17% 이상 성능이 좋다는 결과를 보여주고 있다.

Technology Mapping of Sequential Logic for TLU-Type FPGAs

Jang-Hyun Park[†] · Bo-Gwan Kim^{††}

ABSTRACT

The logic synthesis systems for table look up(TLU) type field programmable gate arrays(FPGAs) have so far studied mostly the combinational logic problem. This paper presents for mapping a sequential circuit onto a popular table look up architecture, the Xilinx 3090 architecture. In the first for solving this problem, combinational and sequential elements which have 6 or 7 input variables are mapped onto one-output CLB's using preprocessing system, and then 6 or 7 input combinational and sequential elements are mapped onto two-output CLB's using sequential mergeability conditions. Finally we assigned the combinational and sequential elements which have less than or equal to 5 inputs. We heavily use the combinational synthesis techniques to solve the sequential synthesis problem. Our synthesis approach is very simple, but its results are reasonable. We compare several benchmark examples with sis-pga(map_together and map_separate) synthesis system and the experimental results show that our synthesis system is 17% better than sis-pga sequential synthesis system for TLU PGAs.

1. 서 론

FPGA(Field Programmable Gate Array)는 Carter [1]등에 의해 제안된 가장 새로운 ASIC 구조로, 기존

의 MPGA(Mask Programmable Gate Array)와 같이 배선영역을 사이에 둔 로직 셀의 2차원적 배열이나, PLD(Programmable Logic Device)와 같이 로직 셀의 기능과 셀들 간의 상호연결을 사용자가 임의로 결정할 수 있도록 되어있다. 이와 같이 기존의 두 종류 구조를 혼합함으로써 FPGA는 짧은 제작 시간, 저렴한 제작 가격, 제작과 테스트 용이성 및 고집적도 등

† 정 회 원: 한국전자통신연구원 선임연구원

†† 정 회 원: 충남대학교 전자공학과 부교수

논문접수: 1996년 3월 4일, 심사완료: 1996년 4월 9일

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, 블럭 구조에 따라 TLU(Table Look-Up) 형과 MB(Multiplexer-Based) 형으로 분류한다. TLU 형 구조에서 로직의 기본 단위를 CLB (Configurable Logic Block)[2]라고 부른다. TLU 형의 기본 블럭은 K개의 입력변수로 어떤 함수도 구현 가능토록 되어있으며, TLU 구조가 결정되면 K는 항상 고정되어 있다. 이 구조의 전형적인 예는 Xilinx(XC3090)[2] 구조로서 이 경우 $K=5$ 이다. 로직 블럭들간의 연결부는 프로그램 가능한 트랜지스터로 구성되어 있으며, 이들 로직 함수와 로직 블럭간의 상호연결은 내부 정적 메모리에 저장되어 있는 프로그램 데이터에 의해서 결정된다. 여기서 TLU형 FPGA의 기술 매핑(Technology Mapping)이란 주어진 함수를 목표로 하는 CLB로 구현하는 것이다.

지금까지 TLU 구조의 FPGA 기술 매핑에 관한 연구는 주로 조합논리 회로에 중점을 두고 연구하였다 [3, 4, 5, 6]. 본 연구에서는 새로운 ASIC 구조로 최근에 관심을 모으고 있는 TLU 형의 FPGA를 위한 더욱더 성능이 좋은 순차회로 논리합성 시스템의 개발을 목적으로 한다. 아직도 순차회로 논리합성에 대한 연구는 초보 단계이며, TLU형 FPGA를 위한 순차회로 기술매핑을 지원하는 시스템은 아직 sis-pga[7]의 한 버전[8]으로만 발표되고 있다. 순차회로 논리합성은 FSM(Finite State Machine)이 주어지면 해당 비용 함수를 최적화하면서 목표 TLU형 CLB에 순차회로를 매핑하는 것이다. 비용함수는 CLB수의 최소화, 네트 수의 최소화, 조합회로 지연시간의 최소화 등이 있으며 본 시스템의 비용함수는 면적과 동일한 CLB 수의 최소화를 사용한다. 이 비용함수를 이용하여 주어진 회로의 비용을 줄이는 최적의 TLU형 FPGA 기술 매핑을 구하고자 한다. 이 과정의 수행은 두 단계로 나누어 질 수 있다. 첫번째는 FSM의 각 상태마다 유일한 코드를 할당하는 상태할당이고, 두번째는 첫번째 단계에서 얻어진 회로를 최적화하고 최적화한 회로를 목표 CLB에 매핑시키는 것이다. 전체적인 순차회로 논리 합성 시스템 SESYS(Sequential SYnthesis System) 개발을 위해서는 순차회로 기술매핑 시스템

이 반드시 필요하므로, 본 논문에서는 먼저 SESYS의 한 부분으로 순차회로 기술 매핑 시스템에 관하여 기술한다. 또한 개발된 본 시스템의 성능을 기존의 sis-pga 시스템과 비교하였다.

순차회로 기술 매핑 시스템 개발을 위한 논문 구성 순서는 지금까지 설명한 서론에 이어 1.1절에서 TLU형 FPGA 구조의 대표적인 Xilinx 3090 구조를 살펴보고, 1.2절에서 순차회로 논리 합성의 문제에 대해서 서술한다. 순차회로 논리합성에서도 조합회로 논리합성이 매우 중요하므로 2.1절에서 조합 논리회로 합성 기술에 대해서 간단히 설명하고, 2.2절에서 순차회로 로직 합성에 대해서 설명한다. 순차회로 기술 매핑을 위한 제안된 알고리즘이 3장에 기술되며, 4장에서는 제안된 알고리즘의 실험결과를 보여준다. 마지막 5장에서는 결론을 내린다.

1.1 CLB의 구조

CLB의 어레이 구조는 사용자 로직을 구현 가능하도록 기능 요소들을 제공한다. (그림 1)에 Xilinx 3090 CLB[2]의 구조를 보여주고 있으며, 각 CLB는 조합회로 부분과 2개의 풀립플롭, 내부 제어 부분으로 구성되어 있다. 입력 신호는 5개의 변수 입력과 한개의 데이터 입력이 있으며, 또한 CLB 외부로 연결되는 2개의 출력신호를 갖고 있다. (그림 1)에 나타난 Xilinx 3090의 CLB 주요 기능은 다음과 같다.

1) 조합 논리 함수

i) 7개의 입력 변수(a, b, c, d, e, Qx, Qy) 중 5개를 입력으로 하는 어떤 함수(F 혹은 G)도 구현이 가능하다.

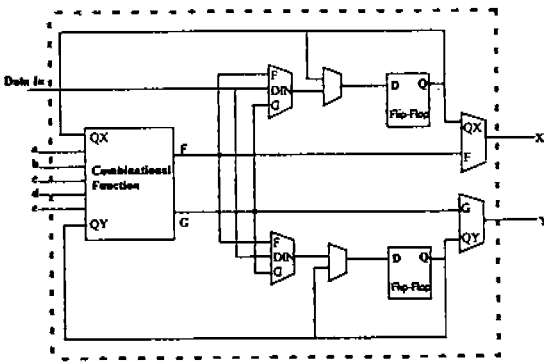
ii) 각 4개 이하의 변수를 입력으로 하는 두 조합 로직 함수(F와 G)를 구현할 수 있다. 이때 조건은 한 입력 변수(a)는 두 함수에 공통으로 들어가고, 두 번째 변수는 b, Qx, Qy 중 하나이고, 세번째는 c, Qx, Qy 중 하나이고, 마지막 네번째 변수는 d 혹은 e 중에 하나이다.

iii) 어떤 함수 F는 6개 혹은 7개의 입력을 가질 수 있다. 여기서 변수 e는 4개의 입력을 가지는 두 함수의 선택변수이다. a와 d는 공통 입력이고, 세번째 변수는 b, Qx, Qy 중 하나이고, 네번째는 c, Qx, Qy 중 하나이다.

2) 순차 논리 회로 구현을 위하여 Qx와 Qy의 출력

을 가지는 2개의 플립플롭을 가지고 있다. 각 플립플롭의 데이터 입력으로는 조합함수의 출력 F, G, 혹은 CLB 외부에서 직접 입력되는 데이터(Din)를 사용할 수 있다.

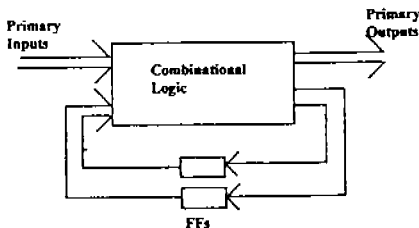
3) 각 CLB는 2개의 출력(X와 Y)을 가진다. 출력 X는 Qx 혹은 F 중 하나이고, 출력 Y는 Qy 혹은 G 중 하나이다. 각 플립플롭의 출력 Qx, Qy는 한 CLB내의 TLU 입력으로 사용 가능하다. 이 기능이 한 CLB 내에서 순차 회로의 케환 루프를 제공한다.



(그림 1) Xilinx 3090 CLB 구조
(Fig. 1) Xilinx 3090 CLB Architecture

1.2 문제 정의

어떤 로직 함수로서 순차회로가 주어지면, 비용함수(CLB의 갯수)를 최소화하면서 목표 TLU 구조(Xilinx 3090)의 조합 및 순차 로직을 가지는 블록으로 주어진 순차회로를 구현하는 것이다. 현재는 비용함수를 CLB의 갯수 최소화 사용하였지만 최종적으로 지연시간, 에지의 갯수 등을 종합한 함수로 변경하고자 한다.



(그림 2) 동기 순차 회로
(Fig. 2) Synchronous Sequential Circuit

2. 기존 알고리즘 분석

(그림 2)의 동기 순차 회로에서 보듯이 일반적인 회로는 조합 로직과 순차 로직으로 구성되어 있으므로 순차회로 로직 합성 문제도 조합회로 로직 합성 기술[3, 4, 5, 6]을 사용하는 것이 필수적이다. 여기서 조합회로 로직 합성 시스템인 sis-pga[7]에 대해서 간략히 살펴보고, TLU 구조를 위한 순차회로 로직 합성 시스템인 sis-pga의 한 버전[8]에 대해서 알아 본다. 본문에서 사용된 노드란 주어진 함수가 분해과정에서 생기는 곱의 합 형태로 표현되는 함수를 말한다.

2.1 조합 로직 합성 문제

sis-pga[7]에서 사용된 로직 합성방법은 먼저 주어진 회로로부터 K개 이하의 입력을 가지는 노드로 구성된 최적회로를 구한다. 이 회로를 이용하여 기술 매핑 단계에서 한 CLB에 배치할 수 있는 함수 짝의 최대수를 구하는 것이다.

1) K 입력 최적 회로

주어진 함수가 기본 블록을 사용하여 K개의 입력으로 구현 불가능한 회로를 구현 가능한 회로로 만들기 위하여 다양한 함수 분해 방법[3, 6, 7]을 사용한다. 이렇게 K개의 입력으로 구현 가능한 회로를 얻은 후 K개의 입력을 유지하면서 사용된 블록수(CLB 수)를 최소화한다.

2) 조합 회로 합병 조건(CMC)

Xilinx 3090 CLB[2]구조에서는 한 CLB에 5개를 입력으로 하는 어떤 함수(F 혹은 G)도 구현이 가능할 뿐만 아니라 어떤 조건만 만족하면 각 4개의 변수를 입력으로 하는 두 조합 로직 함수 F와 G를 구현할 수 있다. 조합회로인 경우 플립플롭의 출력 Qx, Qy를 TLU의 입력에서 무시할 수 있으므로 두 함수 F와 G의 합병 조건은 다음과 같이 간략화 된다. 각 함수는 최대한 4개의 입력을 가지며 두 함수의 입력 변수 합계가 최대 5개이어야 한다. 이것을 조합회로 합병 조건 CMC(Combinational Merge ability Conditions)라 부르며, 주어진 문제는 CMC 조건을 만족하면서 K개 입력 최적회로를 목표 CLB에 구현하기 위한 최대의 함수 짝을 구하는 것이다.

2.2 순차 회로 로직 합성 문제

TLU 구조를 위한 순차회로 기술 매핑 문제를 다룬 시스템은 sis-pga의 한 버전[8]뿐이다. 이 시스템은 먼저 K개 입력 최적 회로를 구현후 기본 블록에서 주어지는 조건을 만족하면서 목표 CLB에 매핑한다. 여기에서 sis-pga는 map_together와 map_separate의 2가지 접근 방법을 사용한다. map_together는 조합회로와 순차회로를 동시에 매핑하며, map_separate는 조합회로와 순차회로를 분리하여 매핑한다. map_together는 주어진 여러가지 CLB 매칭 그래프로부터 모든 할당 가능한 그래프를 찾은 후 카버링 문제 해결을 사용하여 최소의 부분 집합을 선택한다. map_separate는 먼저 조합회로를 CLB에 매핑하고, 다음에 순차회로의 플립플롭들을 CLB에 할당한다.

3. 제안된 매핑 알고리즘

이론적으로 sis-pga의 map_together의해서 계산되어지는 것이 최적의 해를 구하지만 이것은 계산적으로 불가능한 일이다. 더불어 합병 가능한 수와 양방향 매트릭스의 크기가 너무 크기 때문에 휴리스틱 알고리즘을 사용하여도 많은 시간이 요구된다. 이러한 상황에서 속도를 높이며 성능을 향상할 수 있는 방법은 전처리기를 사용하여 먼저 7개의 입력 변수를 갖고 있는 합병 가능한 함수를 선택하고; 그 다음 6개의 입력을, 그리고 나머지는 순차회로 기술매핑 알고리즘을 사용하여 CLB에 매핑한다. 이 알고리즘은 먼저 우선 순위를 가지고 있는 순차회로 합병 조건을 사용한다. 또한 sis-pga의 순차회로 합성 시스템에서는 6개 혹은 7개의 입력과 한 출력을 가지는 노드의 CLB 매핑이 불가능하다. 이유는 K=5인 노드로부터 기술매핑이 시작되므로 6개 이상변수를 갖는 노드를 구성할 수 없기 때문이다.

3.1 순차회로 함수 분해(Decomposition)

서론에서 설명한 바와 같이 CLB의 논리 함수 기능 중 어떤 함수 F는 6개 혹은 7개의 입력을 가질 수 있다. 여기서 입력변수 e는 4개의 입력을 가지는 두 함수의 선택 변수이다. 입력 a와 d는 공통 입력이고, 세 번째 변수는 b, Qx, Qy 중 하나이고, 네 번째는 c, Qx, Qy 중 하나이다. 이러한 기능을 조합 논리 회로 분해

와 sis-pga의 순차회로 합성 시스템에서는 사용하지 않았다. 순차회로 논리 합성에 대해서 이 기능을 사용하면 보다 좋은 결과를 얻을 수 있다. 본 순차회로 기술 매핑 시스템에서는 전처리기로써 이 기능을 첨가 하였다. 먼저 주어진 회로에서 7개 이하 변수를 가지는 노드로 구성된 최적 회로를 구한다. 이 노드 중에서 6개 혹은 7개의 변수로 한 CLB에 구현 가능한 노드를 선택하여 표시한다. 선택 안된 나머지 노드에 대해서 5개 이하의 변수를 가지는 노드로 변환한 후 순차회로 합병 알고리즘을 적용한다.

3.2 순차 회로 합병 조건(SMC)

CLB의 기능에서 각 4개의 변수를 입력으로 하는 두 조합 로직 함수 F와 G를 구현할 수 있다. 이때 조건은 한 입력 변수(a)는 두 함수에 공통으로 들어가고, 두 번째 변수는 b, Qx, Qy 중 선택하고, 세 번째는 c, Qx, Qy 중 하나이고, 마지막 네 번째 변수는 d 혹은 e 중에 하나이다. 이러한 조건을 순차회로 합병 조건 SMC(Sequential Mergeability Conditions)라 부르며 4개의 입력을 가지는 두함수의 최대 입력 변수는 7개 이하이다. 이것이 5개 이하의 입력 변수를 갖는 조합 회로 합병 조건 CMC와 다른 점이다.

3.3 합병 순서

제일 먼저 모든 노드가 7개 혹은 6개의 입력 변수로 합병이 가능한가를 조사한다. 각 노드별로 가중치를 이용하여 7개의 입력 변수로 합병 가능한 노드를 최고의 가중치를 주고 그 다음 6개의 입력을, 마지막으로 5개의 입력을 갖는 노드는 최하의 가중치로 처리한다. 이러한 순차회로 합병 알고리즘은 (그림 3)에 보여준다. 합병 알고리즘의 각 단계별 기능을 설명하면 아래와 같다.

1: SMC를 조사하기 위한 플립플롭의 출력 테이블을 만든다.

2: 한 CLB에 7개 혹은 6개의 입출력 변수가 합병 가능한지 조사한다.

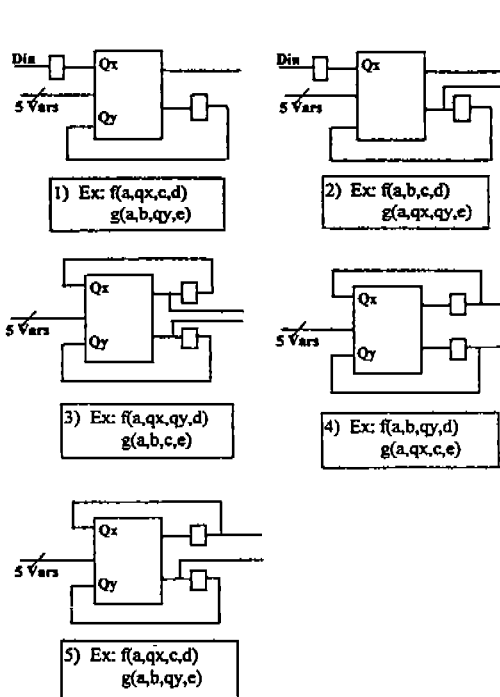
3: 7개의 입력 변수로 합병 가능한 노드들을 최고의 가중치로 선택한다. (그림 4)에 7개의 입력 가능한 패턴 그래프를 보여준다. 이때 한 CLB는 2개의 출력과 7개의 입력 변수(a, b, c, d, e, Qx, Qy)를 가진다.

(1) 경우 1: F와 G가 4개의 입력을 가진 경우

- 한개의 공통 변수
- 데이터 직접 입력(DIN)을 가질 수 있으며 2개의 플립플롭 사용
- (그림 4)의 패턴 그래프와 반드시 일치하여야 한다.
- (2) 경우 2:4개와 3개의 입력을 가지는 경우
- 데이터 직접 입력(DIN)을 가질 수 있으며 2개의 플립플롭 사용
- (그림 4)의 패턴 그래프와 반드시 일치 하여야 한다.

- 1: Make FF_output_table For each node of the network
- 2: check input vars with FF_output_table
- 3: if (node can be merged with 7 inputs) put 7_input_table
- 4: else if (node can be merged with 6 inputs) put 6_input_table
- 5: else put normal_table end_for
- 6: Select the pairing functions
- 7: Check all FF inputs and outputs

(그림 3) 순차회로 합병 알고리즘
(Fig. 3) Merging Algorithm for Sequential Logic



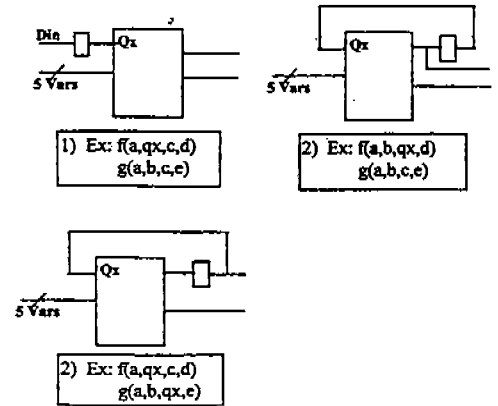
(그림 4) 7개 입력 변수를 가지는 SMC
(Fig. 4) SMCs with 7 input variables

4: 6개의 입력 변수로 합병 가능한 노드들을 가중치로 선택한다. (그림 4)의 경우 2개의 플립플롭을 사용하여 6개의 변수 입력 가능한 패턴 그래프를 보여주고, (그림 5)에는 한개의 플립플롭을 사용한 6개의 입력 가능한 패턴 그래프를 보여준다. 이 때 한 CLB는 2개의 출력과 6개의 입력 변수(a, b, c, d, e, Qx, Qy)를 가진다. 이 경우 함수 F와 G는 각각 4개의 입력 변수, 4개와 3개, 3개와 3개, 4개와 2개 등 여러가지 경우가 있을 수 있다.

5: 이 루틴은 조합회로 합병 조건과 동일하나, 각 함수의 입력 변수가 Qx나 Qy를 가질 수 있다.

6: 각 함수가 가중치로 선택되고 선택되지 못한 함수는 마지막 단계에서 최대 짝짓기 문제를 풀므로써 가장 많은 합병 함수를 구한다.

7: 마지막 단계에서 회로의 모든 플립플롭의 입력과 출력이 정확하게 할당되었는지 조사한다. 만약에 어떤 플립플롭이 CLB에 할당되지 않았으면 새로운 CLB를 추가하여 나머지 플립플롭을 지정한다.



(그림 5) 6개 입력 변수를 가지는 SMC
(Fig. 5) SMCs with 6 input variables

위에서 설명한 새로운 알고리즘은 TLU 구조를 위한 순차회로 로직 합성 시스템 SESYS (SEquential SYnthesis System)의 기술 매핑 시스템으로 구현하였다.

4. 실험 결과

본 논문에서는 SESYS의 TLU형 FPGA를 위한 순

차 논리 기술 매핑 시스템에 대해서 설명했다. 이 시스템의 성능을 분석하기 위하여 K=4와 5인 2가지 종류에 대해서 시험을 수행했으며, 그 결과를 <표 1>과 <표 2>에 보여 주고 있다. <표 1>과 <표 2>의 실험 결과는 아래와 같은 절차를 거쳐서 얻은 것이다. 모든 경우 벤치 마크 예제로부터 jedi[9] 틀을 이용하여 최소 코드 길이 모드로 FSM의 상태를 할당했다. 그 다음 2단계 로직 최적화 프로그램인 espresso[10]를 수행했다. 다단계 로직 최적화 도구인 SIS의 script, rugged[7, 8] 화일을 2번 수행했다. 그 다음 K=4 혹은 5인 회로를 얻기 위하여 TLU을 위한 기술 매핑 script [7]을 수행했다. 이 회로를 구한 후 제안된 순차회로 기술 매핑 시스템의 과정을 수행했다. <표 1>과 <표

2>에서 각 FSM 예제의 입력수(ni), 출력수(no), 상태수(ns), 상태 코딩 비트 수(nb), 조합회로 노드 수, CLB 수 등을 보여주고 있다. 각 K=4와 5에 대하여 SESYS와 sis-pga map_together 및 map_separate의 조합 노드 수와 사용된 CLB의 수를 나타낸다. 표에 나타난 조합노드의 수가 매우 중요하다. 이것이 기술 매핑의 시작점이며 또한 최대 CLB의 갯수이기도 하다. sis-pga의 조합 노드 수와 SESYS에서 사용한 조합 노드 수가 다른 것은 다른 script를 사용했기 때문이다. sis-pga의 map_together의 "-" 표시는 메모리가 모자라거나 수행이 한 시간 이상인 것으로 결과를 얻을 수 없는 경우이다. <표 1>과 <표 2>에 표시한 sis-pga관련 데이터는 참고문헌[7]에서 인용한 것이

<표 1> Xilinx 3090 구조를 이용한 결과(K=4)
<Table 1> Results for Xilinx 3090 Architecture (K=4)

Examples	ni/no	ns/nb	SIS-PGA			SESYS	
			Nodes	Together	Separate	Nodes	CLBs
bbtas	2/2	6/3	7	6	4	6	3
bbara	4/2	10/2	18	11	12	19	11
bbsse	7/7	16/4	37	21	26	36	20
beecount	3/4	7/3	16	10	10	14	8
ex1	9/19	20/5	78	46	52	76	43
ex2	2/2	19/5	42	24	29	38	21
ex3	2/2	10/4	18	11	12	19	10
dk14	3/5	7/3	32	18	19	25	15
dk16	2/3	27/5	82	47	60	79	43
kirkman	12/6	16/4	83	53	53	54	33
keyb	7/2	19/5	68	40	46	72	38
sl	8/6	20/5	75	46	54	53	33
planet	7/19	48/6	186	-	136	185	110
sand	11/9	32/5	167	-	123	177	107
styr	9/10	30/5	142	-	104	149	97
tbk	6/3	32/5	68	50	49	66	43
s386	7/7	13/4	40	23	25	34	23
s510	19/7	47/6	109	-	74	107	64
s820	18/19	25/5	95	57	62	94	54
s832	18/19	25/5	92	67	59	107	59
		Subtotal		530	572		457
	Total				1009		835

<표 2> Xilinx 3090 구조를 이용한 결과(K=5)
<Table 2> Results for Xilinx 3090 Architecture (K=5)

Examples	ni/no	ns/nb	SIS-PGA			SESYS	
			Nodes	Together	Separate	Nodes	CLBs
bbtas	2/2	6/3	5	5	4	5	4
bbara	4/2	10/2	15	13	12	12	11
bbsse	7/7	16/4	29	25	25	25	23
beecount	3/4	7/3	12	11	10	11	10
ex1	9/19	20/5	65	49	48	63	51
ex2	2/2	19/5	33	30	29	25	23
ex3	2/2	10/4	11	10	11	12	11
dk14	3/5	7/3	20	19	18	18	17
dk16	2/3	27/5	49	47	48	49	37
kirkman	12/6	16/4	69	63	63	38	35
keyb	7/2	19/5	54	47	45	54	48
sl	8/6	20/5	55	51	50	39	35
planet	7/19	48/6	144	125	123	143	127
sand	11/9	32/5	133	114	111	137	122
styr	9/10	30/5	113	98	92	116	104
tbk	6/3	32/5	55	49	44	52	45
s386	7/7	13/4	30	26	26	27	25
s510	19/7	47/6	84	73	74	82	75
s820	18/9	25/5	77	68	64	76	61
s830	18/19	25/5	74	61	59	82	70
	Total			985	956		933

다. <표 1>의 제일 마지막 열의 합계(Total)는 K=4인 경우의 map_together “-” 표시를 제외한 모든 결과의 합으로 표시하며, <표 1>의 아래에서 두번째 열의 부분 합계(Subtotal)는 K=4인 경우의 얻을 수 있는 결과를 합계해서 보여주는 것이다. <표 1>에서 보듯이 K=4인 경우 SESYS가 제일 좋은 결과를 보여주고 있다. K=4인 경우 SESYS가 sis-pga의 map_together보다 CLB의 갯수면에서 17% 이상 향상되었다. K=5인 경우 SESYS가 map_separate보다 조금 좋았다. SESYS를 사용하여 K=4인 경우와 5인 경우를 비교했을 때 K=4인 경우가 5인 경우보다 20% 향상된 결과를 보였다. 이유는 K=5인 경우 전체 회로의 노드 수가 적으며 한 CLB에 들어갈 짝을 찾는 데 기회가 훨씬 적기 때문이다. 결론적으로 K=4인 경우가 최적의 기술매핑 결과를 얻을 수 있었다.

5. 결 론

본 연구에서는 새로운 ASIC 구조로 최근에 관심을 모으고 있는 TLU 형의 FPGA를 위한 순차논리 합성 시스템(SESYS)의 기술 매핑 방법에 대해서 설명하였다. SESYS의 순차회로 기술 매핑 방법은 간단한 방법으로 조합논리 회로에서 사용하지 않았던 6개 혹은 7개 입력변수를 갖는 노드 증가 방법을 적용한 후에 가중치에 의한 순차회로 노드 매핑 방법을 사용하였다. 논리 설계 합성기를 개발하면서 최적화 대상으로는 면적, 배선 복잡도, 지연시간 등이 있으나 먼저 면적을 고려하기 위한 비용함수를 만들었으며, 이 비용함수를 이용한 성능 비교 결과 제안된 기술 매핑 방법이 매우 효과적이며, 4장의 실험결과에서 보듯이 CLB 갯수에서 SIS-pga 보다 약 17%의 비용 감소를 이룰 수 있었다.

TLU형의 FPGA를 위한 전체 순차회로 논리 합성의 첫번째 단계로 기술 매핑 방법을 연구하였으나, 앞으로 더욱 복잡한 기능을 첨가 중에 있으며, 지금까지 연구된 이 방법을 후반부 기술 매핑 도구로 사용하여 전반부 도구들을 개발할 예정이다. 전반부 도구로서는 다양한 상태 할당 방법을 가지는 상태 할당 시스템과 TLU 구조에 적합한 조합 및 순차 로직 다단계 함수 분해 도구 등이다. 현재 순차회로 기술 매핑 시스템은 면적의 비용인 CLB 갯수의 최소화에만

중점을 두었다. 회로의 일부 변경이 CLB 및 네트의 수, 즉 면적에 미치는 영향은 상대적으로 국지적이고 비교적 쉽게 정량적 판단이 가능하나, 임계경로의 지연시간에 미치는 영향은 간단하지 않다. 그렇지만, 고성능을 요구하는 최근의 설계요구조건을 고려할 때 지연시간을 고려한 설계 기법의 개발이 필요하다. 앞으로 비용함수를 연장하여 배선 네트의수, 지연 시간 등을 고려한 종합 비용함수로 확장 중에 있다. 추가하여 구현될 시스템은 jedi[9] 대신에 사용할 조합회로 합성에 적당한 상태 할당 시스템을 연구중에 있다. 또한 전처리기로서 사용된 조합회로 분해 기능을 확장하여 TLU형 FPGA에 적합한 다단계 함수 분해 방법을 시험 중에 있다. 이상의 내용들이 완전 개발되면 조합 및 순차회로 로직 합성 시스템인 SESYS를 완성하는 것이다.

참 고 문 헌

- [1] W. Carter et. al., "A User Programmable Reconfigurable Gate Array," Proc. 1986 CICC, pp. 233-235, May 1986.
- [2] Xilinx Inc., 2069 Hamilton Ave. San Jose, CA-95125, "The Programmable Gate Array Data Book", 1994
- [3] R. Murgai, Y. Nishizaki, N. Shenoy, R. K. Brayton and A. Sangiovanni-Vincentelli, "Login Synthesis for Programmable Gate Arrays", Proc. 27th DAC, pp. 620-625, June 1990
- [4] K. Karplus "Xmap: A Technology Mapper for Table-Look up Field Programmable Gate Arrays", Proc. 28th DAC, pp. 240-243, June 1991
- [5] R. Francis, J. Rose, and Z. Vranesic, "Chortle-crf : Fast Technology Mapping Lookup Table-based FPGAs", Proc. 28th DAC, pp. 227-233, June 1991
- [6] R. Murgai, N. Shenoy, R. K. Brayton and A. Sangiovanni-Vincentelli, "Improved Login Synthesis Algorithms for Table Look Up Architectures", Proc. ICCAD, pp. 564-567, Nov 1991
- [7] E. M. Sentovich, R.K. Brayton, and A. Sangiovanni-Vincentelli, et al, "SIS: A System for

Sequential Circuit Synthesis”, Dept. of EECS, Univ. of California, Bekeley, Memo No. UCB/ERL M92/41

- [8] R. Murgai, R.K. Brayton and A. Sangiovanni-Vincentelli, “Sequential Synthesis for Table Look Up Programmable Gate Arrays”, Proc. 30th DAC, pp224-229, June 1993
- [9] B. Lin and A.R. Newton, “Synthesis of Multiple Level Logic from Symbolic High-Level Description Languages”, Proc. of International Conf. on VLSI, 1989
- [10] R.K. Brayton G.D. Hachtel, C.T. McMullen and A. Sangiovanni-Vincentelli, “Logic Minimization Algorithms for VLSI Synthesis”, KAP, 1985



김 보 관

1976년 서울대학교 전자공학과 (공학사)
 1978년 한국과학기술원 전기 및 전자공학과(공학석사)
 1989년 Univ. of Wisconsin Madison 전기 및 컴퓨터 공학과 (공학박사)

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



박 장 현

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

주관심 분야: 설계자동화, 개인통신(PCS)