

---

# 하드웨어 설계 교육에서의 TOP-DOWN 접근방법

## : 논리설계 과목을 중심으로

이강, 정경훈, 한윤식

한동대학교 전산전자공학부

(2003. 10. 13. 접수)

### A Top-Down Approach to the Hardware Design Education Focusing on the Logic Design Courses.

Kang Yi, Kyeong-Hoon Jung, Youn-Sik Han,

*School of Computer Science and Electronic Engineering, Handong Global University*

(received October 13, 2003)

#### 국문요약

하드웨어 설계 교육의 궁극적인 목표는 학생들로 하여금 시스템 설계능력을 갖추도록 배양하는 데에 있다. 그러나 상당수의 기존 교과과정은 설계 자체보다는 개별적인 빌딩 블록의 내부동작 원리의 이해를 중심으로 운영되는 것이 현실이다. 따라서, 학생들의 학습동기 유발에 실패할 뿐만 아니라 정작 현장에서 필요로 하는 시스템적 관점에서의 설계를 충분히 경험하지 못하는 것이 문제점으로 지적된다. 이러한 문제점들을 극복하기 위한 대안으로, 과목의 초기 도입부에 추상화의 수준을 높여서 시스템의 전체 설계를 조망할 수 있도록 하고, 점차로 추상화의 단계를 낮추어서 결국 소자의 동작 원리를 탐구하는데 이르도록 강의의 순서를 역전시킬 필요성이 있다. 본 논문에서는 하드웨어 설계의 입문 과목인 논리설계 교과목을 대상으로 하여, 시스템 수준의 설계표현 및 검증방식을 먼저 소개하고 소자에 대한 세부지식은 후반부에 소개하는 새로운 교육방식을 제안하고 이의 타당성을 검증하기 위해 교내의 동아리를 활용한 비정규 교육과정에 이를 도입한 사례를 소개한다.

#### Abstract

The ultimate goal of a hardware design course is to equip the students with the system design ability. However, the majority of the current structures of the design courses are focused on the understanding of the operational principles of each device which is used later as a building block for the design of a system. The shortcomings of this approach are, first, that it is very hard to keep the students motivated to the end of the course where system design concepts are dealt, and,

second, the students do not have enough experience of the system design which is usually required in the field. As an alternative to solve these problems, it is necessary to reverse the order of contents of the course. Namely we introduce the high level of the abstract concept of the system design in the very beginning of the course and later by lowering the level of abstraction to the operational principle of the internal devices. In this paper, we propose a new top-down methodology for the introductory hardware design course of logic design, where the design expression and verification in the system-level are introduced first and then detail knowledge on each device is introduced later. Also, we report a case result from a student's working group as part of an extracurricular education in order to verify the validity of our proposed approach

## I. 서 론

공학 교육에 있어서 실효성과 수강자들에 대한 동기 유발은 교육 내용 그 자체 못지 않게 중요하다고 할 수 있다. 기존의 하드웨어 설계관련 교과목들의 강의 방식이 이런 측면에서 만족스럽지 못하다는 판단에 따라 새로운 교육 방식을 논리설계 과목을 중심으로 제안하고자 한다.

컴퓨터공학 교육은 크게 하드웨어와 소프트웨어 분야로 나뉘어진다. 그러나, 다음 3가지 이유로 인하여 소프트웨어 또는 하드웨어 한쪽에만 국한된 교육이 아니라 양자에 대한 균형잡힌 교육을 필요로 한다. 첫째 내장형 시스템의 중요성이 부각되면서 하드웨어와 소프트웨어 양자에 대한 이해와 최소한의 구현 능력이 필요하기 때문이다. 내장형 시스템에서의 소프트웨어 개발은 PC 환경과는 달리 표준화된 하드웨어 환경이 아닌 특정 하드웨어 플랫폼에서의 프로그램 개발이므로 하드웨어에 대한 깊이 있는 이해와 소프트웨어와 하드웨어와의 상호작용이 더욱 긴밀하게 요구되기 때문이다. 둘째로, 하드웨어 개발자와 소프트웨어 개발자는 모두 상대 영역에 대한 전문지식이 있을 때 더욱 효율적인 시스템의 구현이 가능하기 때문이다. 셋째, 하드웨어와 소프트웨어의 통합 설계가 주요 흐름으로 자리잡고 있기 때문이다.

이와 같이 하드웨어 설계 교육이 중요하게 됨에 의해서 이 분야의 기초 과목인 논리 설계 과목(또는 디지털 논리, 논리 회로, 디지털 시스템, 디지털 공학) 교육은 하드웨어 설계 전문가를 물론

이고 소프트웨어 전문가에게도 필수적이다. 논리 설계 과목은 거의 모든 컴퓨터 및 전산 관련 학과와 전자 분야의 학과에서도 오랫동안 정규 필수 과목으로 자리잡아 왔다.

논리설계가 정규 과목으로 자리잡은 시점으로부터 지금까지 많은 여건의 변화가 있었다. 설계자동화 도구의 성능 개선과 보편화, 회로 구현 소자의 고집적성과 유연성 확대 등으로 하드웨어 설계 환경이 놀라운 속도로 변화되었고 설계자에게 요구되는 자질도 많이 변화되었다. 더욱이 소프트웨어와 하드웨어 간의 긴밀성도 더욱 높아지는 등 설계 대상 시스템의 구성과 관심의 중심도 달라졌다. 따라서, 수 십년간 지속된 하드웨어 설계의 입문 과목으로서 논리설계 과목이 앞으로도 제 역할을 충실히 해나가기 위해서 그 운영 방식과 강의의 내용에 있어서 재점검할 시점에 와있다고 판단된다.

본 논문에서는 새로운 시대적 요구사항과 학습의 효율성을 동시에 수용하기 위한 논리설계 과목의 교육방법의 재구성을 제안한다. 하드웨어 설계에서의 시스템적 조망을 제공하고, 학습 동기의 조기 유발을 가능하게 하고, 현장의 실용적인 필요에 부응하는 논리설계 교육이 되기 위해서 과목 내용을 재배열하고, 강의 내용의 중심을 소자 원리 설명과 이론에서 설계 자체로 이동하고, 새로운 설계 환경을 강의 내용에 수용하고 이를 충분히 활용 하기 위한 구체적 방법들을 제안한다.

다음 장에서는 논리설계 교과목을 중심으로 하여 기존의 bottom-up 방식의 교육방식의 문제점

을 살펴보고, III장에서 이의 대안으로써 top-down 방식을 제안한다. 그리고 IV장에서는 새로운 교육방식의 적용 사례 및 성과를 제시하고 마지막으로 V장에서 결론과 향후 과제들을 제시한다.

## II. 기존의 bottom-up 교육 방식 : 논리설계를 중심으로

하드웨어 설계와 관련된 대부분 교과목의 학습 방법은 전통적으로 bottom-up 방식으로 이루어지고 있다. 즉 설계를 위한 기초 배경이 되는 지식을 학습하고, 이어서 설계의 기본단위가 되는 빌딩 블록들을 하나씩 다루고 난 후, 교과목의 성격 및 형편에 따라 시스템 설계를 배우는 방식이 일반적이다.

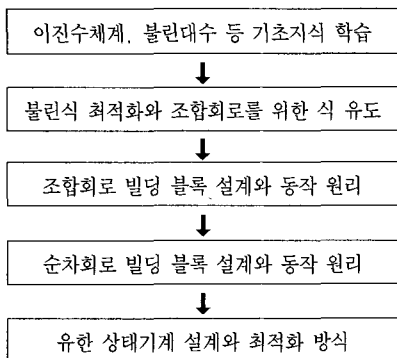
논리설계의 경우를 예로 들어 생각하면, 다음의 <그림 1>과 같이 기초 배경 지식이 되는 이진수 체계 및 불린대수(Boolean Algebra)의 학습에서부터 시작하여 조합회로, 순차회로 그리고 유한상태기계(Finite State Machine)를 다루는 단계를 거친다.

이러한 bottom-up 접근방식은 지금까지 상당한 교육효과를 보여왔지만, 현재의 입장에서 보면 한편으로는 적지않은 문제점을 나타낸 것도 사실이다. 우선 원리위주의 교육방식이 학생들의 학습 동기 유발에 실패하였다는 점을 들 수 있다. 이러

한 학습과정을 따르게 되면 우선 불린 대수를 배우고 논리 게이트 수준의 소자들의 동작 원리 설명과 논리 최적화 등의 수학적 이론 설명에 한학기의 절반 이상의 시간을 할애하게 되고 정작 순차회로와 제어부 설계에 필요한 유한상태기계의 설계에 대한 것은 학기의 후반부에 소개된다. 따라서 시간적인 제약으로 인해 시스템의 설계 자체에 필요한 안목을 가질 기회보다는 개별적 상태도(State Diagram) 하나 하나의 구현을 위한 불린식 유도에만 집중하다가 수업을 마쳐야 하는 경우가 발생하기 쉽다. 이에 따라 학생들은 학습 초반에 자신들이 배우는 내용의 필요성과 전체적 안목을 가지지 못한 채, 이론에 치우친 개별적인 세부 내용에 대한 학습에 치중하다가 정작 중요한 시스템 설계 단위에서는 학기말이 되면서 다른 과목의 학습부담과 겹치면서 자연스럽게 학습의욕이 저하되는 문제점이 나타난다.

이러한 문제는 한편으로는 지금까지의 논리설계 교육에 주로 사용된 교재에 기인한 바가 크다. 예를 들어 가장 널리 알려진 논리설계 교재 (M. Mano, Logic and Computer Design Fundamentals; M. Mano, Digital Design ; R. Katz, Contemporary Logic Design ; S Brown et al, Fundamentals of Digital Logic with VHDL Design)들의 구성과 내용을 살펴보면 다음 <표 1>과 같다. <표 1>에서 Mano1은 (Mano, Logic and Computer Design Fundamental)를 의미하고, Mano2는 (Mano, Digital Design)을 의미한다.

이들 교재에서 살펴볼 수 있는바, 교재의 내용을 따라 수업을 진행한다면 학기말에 가서야 학습의 최종적인 목표인 시스템 설계를 다루게 되는데 이때에는 이미 다수 학생들이 지루한 논리식의 추출과정과 기본이 되는 빌딩 블록들의 내부 회로 동작 설명을 따라가기에 지쳐서 시간적으로나 정신적으로 여력이 얼마 남지 않은 상태이다. 따라서 파편적 지식만 배운 채 학기를 마칠 위험성이 있다. 즉 기본 빌딩 블록을 중심으로 교육하다가 학습분량이 지나치게 늘어나서 실제적으로 설계를 해볼 시간적인 여유가 부족하다는 문제점이



<그림 1> 기존의 논리설계 교육 방식

〈표 1〉 기존의 대표적인 논리 설계 과목을 위한 교재들의 목차

주제	Mano1	Mano 2	Katz	Brown
기초배경	1장	1장~2장	1장	1장~2장
조합회로	2장 ~ 3장	3장~4장	2장~3장, 5장	4장~6장
순차회로	4장 ~ 5장	5장~6장, 9장	6장~10장	7장~9장
메모리/PLD	6장	7장, 10장	4장	10장일부
종합적 설계	7장 ~10장	8장, 11장	11장~12장	10장일부

있다.

다음의 〈표 2〉는 논리설계 교과목의 수업내용을 기초배경지식, 조합회로, 순차회로, 메모리 및 PLD, 그리고 종합적 설계로 구분하고 국내의 여러 대학의 강의계획서를 분석하여 해당 내용에 따라 강의분량을 정리한 것이다. 여기에서 살펴볼 수 있듯이 대부분의 대학에서 bottom-up 방식의 수업을 진행함에 따라 종합적 설계에 해당하는 부분의 강의가 거의 이루어지지 않음을 알 수 있다.

한편, 대부분 학교에서는 논리설계 이론교육과 병행하여 논리설계 실험교육을 함께 병행하여 실시하는 경우가 일반적인데, bottom-up 방식의 접근은 실험과 이론의 연계교육을 생각할 때 문제점을 안고 있다. 대부분의 논리설계 실험에서는 전압 레벨의 측정, TTL을 이용한 게이트의 동작 확인, 만능보드(bread board)를 이용한 회로의 구성 등으로 많은 실험 시간들을 보내고 있다. 물론 이러한 실험교육이 전자회로를 설계하고 PCB 보드를 제작하는 데에는 필요한 것은 당연하지만,

실제 현장에서 만나는 복잡한 시스템 레벨에서 설계하는 실습 기회를 가지지 못하는 단점이 있다. 이는 이론 교육에서 학습한 내용을 실험을 통해 확인하는 단계를 거치게 되는데, 앞서 언급했듯이 이론 교육에서 시스템 레벨의 설계를 다루는 것은 학기의 후반부에서나 가능하기 때문이다.

### III. 제안된 top-down 교육 방식 : 논리설계를 중심으로

II장에서는 기존 bottom-up 방식의 학습의 문제점으로 다음의 세가지를 지적하였다. 첫째, 원리위주의 학습은 학생들의 학습동기 유발에 실패하였다. 학생들에게는 자신들이 학습할 내용의 필요성과 전체적인 안목을 가지는 것이 중요하다. 둘째, 기본 빌딩 블록 중심의 교육방식은 많은 학습분량으로 인해 정작 중요한 시스템 설계를 다룰 시간적인 여유가 부족하게 되어 이는 논리설계를 학습했음에도 정작 설계는 하지 못하는 학습목표 달성의 부실로 이어진다. 셋째, 이론교육과 실험

〈표 2〉 국내 4년대 종합대학에서의 논리설계 과목 강의 계획서 분석결과 (주제별 강의 주수)

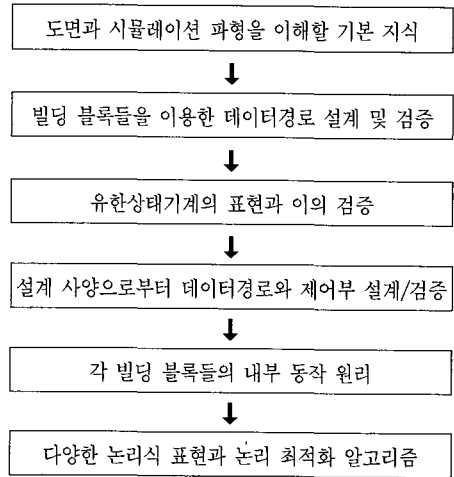
	국립A	국립B	사립C *	사립D	사립E	국립F	국립G	사립H
사용 교재	Mano1	Mano2	Katz	Mano2	Mano2	Brown	Katz	Yarb
기초배경지식	3	2	3	2	3	2	1	3
조합회로	4	6	4	3	4	4	3	4
순차회로	5	5	6	6	5	5	6	4
메모리/PLD	2	1	1	4	2	2	1	2
종합적설계	0	0	0	0	0	0	2	0

\* 이론과 실험이 통합된 과목이나 이 가운데 이론부분만을 분석

교육 사이의 진도차이로 인해 두 교과목의 연계운 영이 쉽지 않게 되고, 이로 인해 마땅히 나타나야 할 상호간의 시너지 효과가 사라질 수 있다.

이러한 문제점을 극복하기 위하여 일단 시스템 설계에 대한 경험을 한 뒤에 그 원리를 나중에 배우는 방식으로 강의의 순서를 재배열할 필요가 있다. 즉, 회로에 대한 추상화(Abstraction)의 단계를 높여서 전체를 조망하고 설계할 수 있는 능력을 부여한 뒤에, 추상화의 단계를 점차로 낮추어 결국 소자의 개별적 동작원리와 설계자동화 도구의 핵심인 최적화 알고리즘의 수학적 배경이론까지 공부한다는 전략이다. 세부적인 빌딩블록 내부의 동작 원리를 모른 채로 시스템을 성공적으로 설계를 할 수 있는 이유는, 매우 발전된 상위단계의 설계 자동화 도구들을 무료로 사용할 수 있는 여건이 FPGA(Field Programmable Gate Array) 공급사들을 중심으로 현재 조성되어 있기 때문이다. 예컨대, Xilinx 사의 홈페이지 (<http://www.xilinx.com>)와 Altera사의 홈페이지 ([www.altera.com](http://www.altera.com))에서 다양한 방식의 논리 회로의 기술(description), 시뮬레이션, 자동 합성, 칩의 레이아웃 생성과 만들어진 칩 이미지의 다운로드에 이르는 모든 설계 및 검증 툴 일체인 ISE Webpack과 Quartus Web Edition를 무료로 다운로드 받아서 사용할 수 있다. 조합회로의 경우, MSI 수준의 빌딩블록들을 포함한 라이브러리가 제공되는 스키매틱 에디터와 논리 시뮬레이터가 있고, 순차회로의 경우, 상태를 그림으로 그려서 동작을 기술할 수 있는 graphic state machine editor와 스키매틱 에디터에서 즉시 사용 가능한 레지스터와 카운터 등의 기본 빌딩블록들이 라이브러리로 제공된다. 이런, 고수준의 설계 자동화 도구들의 도움을 받아서 회로 빌딩블록 내부의 세부 동작 '원리' 보다는 시스템의 요구 사항을 어떻게 '표현'하고 '검증'하는냐에 더 집중하여 수업이 가능하다. <그림 2>는 이러한 원리에 의한 수업 방식을 위한 흐름도를 제시하고 있다.

<그림 2>에 제시된 내용은 다음과 같이 10단계로 나누어 설명할 수 있다.



<그림 2> 새로운 논리설계 교육 방식의 흐름도

- (1) 이진법과 진리표의 이해 및 논리 회로 도면과 시뮬레이션 파형의 이해 (1주)
- (2) 논리 회로 빌딩블록들의 행위 수준(Behavioral Level)의 동작 이해 : 스키매틱 에디터나 HDL에 의해서 MUX, Decoder, Encoder, Adder, Register, Counter 등의 빌딩블록들의 상호연결에 의한 데이터패스 회로의 기술과 설계 자동화 도구를 이용한 검증 (1주)
- (3) 상태도의 이해와 상태를 설계자동화 도구로 기술하고 이를 검증하는 연습 (1주) : 그래픽 도구나 HDL을 이용하여 설계하고 자동합성 도구에 의해서 검증
- (4) 설계 사양에 따라 데이터 패스와 제어부를 나누어 설계하고 검증하는 연습 (2주)
- (5) 플립플롭과 래치 등의 기억소자 동작 원리와 타이밍 이해하기 (2주)
- (6) 레지스터와 카운터 동작 원리 이해하기 (2주)
- (7) 조합회로의 심화 학습 (2주)
- (8) 자동 합성 툴의 내부 알고리즘으로서의 조합회로의 다양한 표현 방식과 최적화 알고리즘 알아보기 (2주)
- (9) 자동합성 툴의 내부 알고리즘으로서의 유한상태기계의 최적화와 상태식(state

equation)의 유도 과정을 학습하기. (2주)  
 (10) PLD, Memory 등에 대한 개별 칩의 동작이해 (1주)

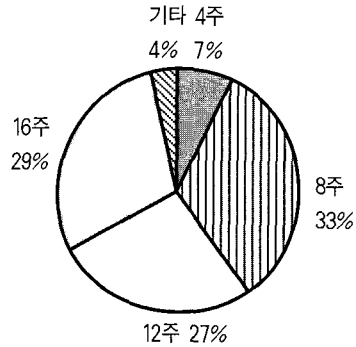
과정 (4)까지를 통하여 시스템을 설계하고 이를 설계 자동화 도구를 사용하여 표현하고 검증하는 과정을 거치면서 시스템의 전체적 설계의 관점을 배우게 된다. 초반에 설계 자체를 배운 뒤 과정 (5)부터는 동기와 흥미가 유발된 상태에서 내부 원리를 배우는 과정이 된다.

제안된 방법에 의해서 이론 과목이 조기에 시스템 설계 방법론과 관점을 배우도록 기회를 제공하게 되면, 논리설계 실험내용도 소자 수준의 '이론'의 확인보다는 '시스템 설계' 자체에 집중하는 것이 가능해진다. 회로의 구현에 있어서도 TTL 소자 칩들에 의존하는 기존 방식에서 벗어나서 MSI급 칩들과 범용 FPGA 보드를 활용하여 배선과 보드 디버깅에 보내는 시간을 최소화하여 실험의 내용을 전자적 원리의 확인보다는 규모가 큰 시스템의 설계에 더 많은 시간을 보내도록 할 것을 제안한다.

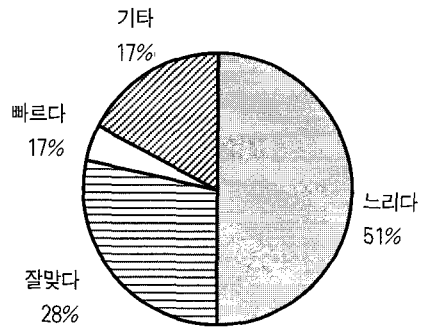
#### IV. 제안된 교육 방식의 적용 사례 및 성과

본 논문의 문제제기가 옳은지를 검증하기 위하여 논리설계 과목을 기존의 방법으로 수강한 50명의 학생들을 대상으로 설문조사를 실시하였다. 다음 <그림 3>과 <그림 4>는 설문 조사한 결과이다. <그림 3>의 질문은 '논리 설계 교과목을 통하여 논리회로를 설계할 수 있는 안목이 생긴 것은 언제부터인가' 이고, <그림 4>의 질문은 '논리설계 이론과목의 진도가 실험과목의 진도에 비해서 어떠한가' 하는 질문이다.

설문조사 결과에서 알 수 있듯이 기존의 논리설계 방식으로는 56%의 학생이 중간시험 이후에 가서야 설계에 대한 안목을 가지게 되었으며, 이론과목의 진도와 실험과목의 진도가 잘 맞거나 실험의 진도가 오히려 빠르다는 응답은 전체의 32%에 불과하였다. 이는 본 논문에서 제기한 문제점에



<그림 3> 설문조사 1의 결과



<그림 4> 설문조사 2의 결과

대한 분석이 타당함을 의미한다.

앞 장에서 제안한 새로운 교육 방식을 한동대학교 전산전자공학 전공 학생들로 이루어진 하드웨어 설계 동아리의 교육에 도입하여 적용하였다. 동아리 교육은 4주간의 여름방학 기간을 이용하여 이루어지는데, 이러한 교육방식을 도입한 결과 상대적으로 짧은 기간임에도 불구하고 중간 정도의 복잡도를 가진 디지털 시스템을 FPGA로 만드는 교육 성과를 거두었다. 대상 학생들은 정규 교과과정 상의 논리설계 과목을 이수하기 전인 2학년 1학기를 마친 학생들로서 본교에서는 무전공으로 입학하여 2학년에 진학하면서 전공을 선택하기 때문에 2학년 여름방학은 사실상 첫 전공 학기를 마친 방학이다. 이 2학년 첫 여름 방학 기간 동안 동아리의 회원이 된 2학년 학생들은 선배들로부터

1주 정도의 집중적인 논리설계 기초와 1주 동안의 VHDL 언어 교육을 받고 남은 2주 기간에 VHDL 언어를 이용하여 FPGA 보드를 이용한 실제 시스템 설계 실습에 들어간다.

교육과정의 전반기에서 배우는 논리설계의 기초는 상세한 소자들의 동작 원리와 회로 최적화를 위한 수학적 이론은 가능한 한 제외한 채, 시스템 설계를 위한 지식을 중심으로 이루어진다. 실제로 회로를 정확히 기술하기만 하면, 논리 최적화는 설계자동화 도구들이 자동적으로 수행하기 때문에 설계에 있어서 핵심적인 내용은 회로의 상위단계에서의 '설계'와 이의 '표현'이기 때문이다. 다음 <표 3>은 동아리에서 운영하는 교육과정에 대한 요약이다.

이상의 적용 사례는 동아리를 대상으로 한 방학동안의 집중적인 과정이라는 점에서 정규교육 과정과는 차이가 없지 않지만 몇가지 측면에서 긍정적인 결과를 얻을 수 있었다. 우선 자발적으로 모인 모임이므로 어느 정도 동기 부여가 되어 있는 상태임을 감안하더라도 이 프로그램에 참여한 대부분의 학생들이 하드웨어 설계분야에 지속적으로 열의와 관심을 가지게 되었고, 2000년 이후 매 방학마다 학생들을 중심으로 하드웨어 설

계 프로젝트가 계속 진행되고 있다. 또한 이 프로그램에 참여한 대부분의 학생이 전공을 1학기 밖에 경험하지 못했고 논리설계 과목조차 전혀 접해보지 못한 2학년 초반의 학생들임에도 불구하고, 최종적인 결과물로서 VHDL을 이용하여 구현한 중간 정도 복잡도의 설계를 완성했다는 것은 제안 방식의 학습효과에 대해 긍정적으로 기대할 수 있다.

## V. 결 론

오늘날 설계자동화 도구의 발전은 세부적인 소자의 동작 원리를 모르고도 시스템 설계를 가능하게 만들었으나, 기존의 하드웨어 설계교육은 원리 설명에 치중하다보니 학생들의 흥미유발 면에도 부족하였고 시스템의 설계 자체에 대한 교육을 충분히 시킬 수가 없었다. 이에 대한 대안으로서 전체적으로 조망할 수 있는 큰 그림을 먼저 제시한 뒤에 내부 원리를 보여주는 방식에서의 교육방식의 전환이 필요하다. 본 논문에서는 논리설계 교과목을 대상으로하여 과목 초반부에 설계자동화들의 사용법과 더불어 시스템 설계에 대한 학습기회를 제공하고 기존의 방식에서 전반부에 다루던

<표 3> 동아리에서 운영중인 방학중 단기 과정의 내용 및 결과

강의주차	주 제	세부 내용	교육 단계
1	논리설계 기초 교육	1일: 논리 회로의 기초 2일: 조합회로 빌딩블록 3일: 순차회로 빌딩블록 동작 4일: 상태도 설계 5일: 시스템 설계 방법론	(1), (2)
2	VHDL 언어교육	문법 자체보다는 하드웨어 합성 중심으로	(3), (4)
3-4	FPGA 보드를 이용한 설계 실습	만능 보드 대신 FPGA 보드 사용	(3), (4), (5), (6), (7)
4	FPGA 보드에서 구현한 최종 실험결과물 목록 예	스톱워치 기능을 가진 디지털 시계 주사위 게임기 4칙연산기 엘리베이터 제어기 교통신호 제어기	(3), (4), (5) (6), (7), (10)

원리 설명을 후반부로 옮겨서 세부적인 알고리즘과 빌딩블록 내부의 동작을 교육하는 새로운 교육 방식을 제안하였다. 이를 동아리의 교육과정에 적용한 결과 학생들에게 시스템에 대한 설계 능력을 배양하고 학습 동기를 유지한다는 측면에서 고무적인 결과를 얻을 수 있었다.

본 제안방식이 가지는 예상되는 문제점으로는 기존 방식에서 초반부에 집중적으로 교육하는 조합회로의 최적화 알고리즘에 대한 학습이 부족해질 우려가 있다. 그러나 이는 설계자동화 툴 제작에 관심이 있는 일부 학생들의 문제로서, 하드웨어 설계의 입문과목으로 논리설계를 생각한다면 대부분의 학생들은 설계 자체가 더 중요한 학습목표라고 할 수 있다. 한편, 여기서 제안하는 방식이 정규교육과정에 적용되기 위해서는 이러한 제안을 충실히 따르는 논리설계 교재의 집필과 이에 따른 시범 정규 교과목 운영이 향후 과제로 남아 있다.

[ 참고 문헌 ]

- M. M. Mano and C. R. Kime, Logic Design and Computer Design Fundamentals, Perentice-Hall
- M. Mano, *Digital Design*, Perentice-Hall
- R. H. Katz, Contemporary Logic Design, The Benjamin/Cummings Publishing Company
- S. Brown and Z. Vranesic, Fundamentals of Digital Logic with VHDL Design, McGraw Hill
- J. M. Yabrough, Digital Logic Applications and Design, PWS Publishing Company
- J. F. Wakerly, Digital Design Principles & Practice, Prentice Hall
- W. Kleitz, Digital Electronics A Practical Approach, Prentice Hall