

Top-Down 방식의 계층적 논리 설계용 Schematic Editor에 관한 연구

○ 김 고 선 김 봉 일 이 분 기  
연 세 대 학 교 전 자 공 학 과

A Study of Schematic Editor for Top-Down Methodology of Hierarchical Logic Design

Kyo - Sun Kim, Bong - Ryul Kim, Moon - Key Lee  
Dept. of Eletronics Engineering, Yonsei Univ.

Abstract

This paper describes a special-purpose schematic editor program which is appropriate for Top-Down methodology of hierarchical logic design. The data structure which has capability to be shared with multi-level logic simulator is developed. For the purpose of efficient search, a new indexing method is designed. That is exponentially distributed one dimensional bin based on probability of presence of graphic elements.

1. 서론

초대형 집적회로 설계의 복잡성이 증가함에 따라 계층적 설계 방법을 필요로 하게 되었다. 각 설계 방식에 따라 적합한 Tool의 도움이 있어야 하며 특히 계층적 논리 설계 방식에 있어 Tool은 설계 환경에 대한 융통성, 효율적인 검증 시스템, 컴퓨터의 기억용량과 소요 시간의 최소화, 그리고 설계자들의 독립성 등의 효용을 가지고 있어야 한다[1]. Top-Down 설계 방식에서 최종설계의 불확실성, 각 Macro Block의 상호 의존성의 문제를 해결할 수 있는 Tool의 성격은 각 Macro block, 그리고 sub-macro block의 모델링이 가능한 다층 논리 검증을 요구한다. 본 논문은 각 모듈의 모델링과 실제 설

계간의 일관성을 유지할 수 있고 multi-level logic simulator 와 database 를 공유할 수 있는 schematic editor 의 구현을 서술한다. graphic은 GKS를 이용하였다.

2. 계층적 논리 설계 시스템

계층적 논리 설계 과정은 functional specification 을 시작으로 하며 그 시스템을 설계하기에 알맞는 크기의 기능 블록이 될 때까지 Macro로 분할한다. 이 기능 블록들은 먼저 software 모델링으로 정의되며 이것은 다시 Boolean logic 블록으로 모델링 된다.

각 Macro 설계에 있어 완성된 Macro는 모델링된 다른 Macro 와 함께 검증할 수 있게 된다. 따라서 설계자가 다른 Macro의 완성을 기다리지 않고 독립적으로 설계 및 검증을 수행할 수 있게 된다.

3. 시스템의 구현

본 시스템은 일반적인 Top-Down 방식의 설계를 할 수 있는 schematic editor로서 설계과정에서 incremental 하게 schematic을 capture할 수 있으며 multi-level simulator module을 부착하면 database를 공유하면서 즉시 simulation 을 할 수 있는 형태의 data 구조를 가지고 있다.

3.1 Data 구조

이 시스템의 data 구조는 그림 1 과 같다.

Circuit Symbol Definition (CSD), Graphic Symbol Definition (GSD), Circuit Symbol Use (CSU), Graphic Symbol Use (GSU), Wire, 그리고 일차원 bin (xBin, yBin) 들로 구성되어 있다. 이때 bin은 GKS 에서 효율상 Segment로 정의하지 않는 primitive들을 효과적으로 첨가, 탐색, 삭제할 수 있도록 마련되어 있다.

- (1) CSD - submodule의 behavioral abstraction을 표현하며 현재 implementation 에서는 입출력에 대한 정의만 갖고 있고 simulation을 위한 정보 (모델링 또는 완성된 Macro)로의 pointer는 disable 되어 있다.
- (2) GSD - Symbol Editor에서 생성된 symbol에 대한 graphic primitive들을 담고 있으며 입출력에 대해 CSD와 mapping 되어 있다.
- (3) CSU - CSD 의 Instance로서 simulation에서 CSD 모델을 이용하도록 되어 있고 현재는 CSU와 연결만 맡고 있다.
- (4) GSU - Schematic Editor에서 GKS의 Segment로 정의되며 GSD의 Instance로서 각 입출력 단자는 Connector를 통하여 Wire를 소유한다. 또한 Symbol의 Graphic primitive들은 GSD의 image로서 memory를 별도로 할당하지 않는다.
- (5) Wire - 하나의 Net를 의미하며 line과 tap을 소유한다. GSU의 Connector에 의해 소유되고 line이나 tap이 추가 또는 제거에 의해 merge 또는 split 된다. GSU의 Connector 에 연결되지 않는 Wire는 따로 묶여 존재하게 된다.
- (6) bin - line과 tap, connector, GSU의 bounding box들을 탐색에 유리하도록 저장하는 구조이다. 모든 line은 manhattan 형식으로 수평 line은 y좌표를 key로 yBin에, 수직 line은 x좌표를 key로 xBin에 속하게 되며 GSU의 bound-

ing box들도 네 개의 변으로 나뉘어 각각 이 bin에 들어 가게된다. 각 bin의 크기는 원소가 존재할 확률을 고려하여 좌표 평면의 외곽으로 나갈수록 지수함수적으로 커지게 되어 전체 bin의 갯수는 크지 않게 된다. (각각 약 100개). 이것은 등간격 bin에서 각 bin에 속하는 원소의 갯수가 확률적으로 많아 지지 않을 것이라는 효율이 있다. (그림 2 참조)

3.2 연결도 검사 Algorithm.

- (1) 수직 line이 추가 되면 그것의 y좌표를 key로 그 line이 지나는 yBin들을 검색하여 GSU의 bounding box와 교차하는가 확인한다.  
만약 교차하면 Symbol위로 line이 지나감을 의미하므로 불가능함을 알리고 끝난다.
- (2) 이 수직 line에 대해 구조에 해당하는 memory를 할당하고 이 line의 x좌표를 key로 xBin에 넣는다.
- (3) 이 line의 양끝의 y좌표에 해당하는 yBin에서 이 y좌표를 가지고 있으면서 이 line과 교차하는 line을 찾아 그 line이 소속된 Wire에 이 line을 첨가한다.
- (4) 만약 양 끝에서 만난 Wire가 다르다면 두 Wire를 merge한다.
- (5) 수평 line 도 같은 방법으로 수행한다.

4. 결론

본 논문은 계층적 논리 설계에 적합한 특수 목적용 schematic 편집 프로그램 개발에 대해 서술하였다. 또한 incremental 한 schematic capture 기능과 효율적인 탐색을 위한 data 구조와 Algorithm에 대해 설명하였다. stimuli 편집 기능과 다층 논리 검증 모듈을 첨가하면 Top-Down방식의 계층적 논리 설계용 Tool의 기능을 모두 보유할 수 있다. 특히 data 구조가 검증 모듈과 공유할 수 있는 형태이므로 사용자가 Tool과 많은 정보

복 동시에 주고 받을 수 있으며 중간 형태의 data storage가 적어져 컴퓨터 이용 효율면에서 유리한 Tool이라 할 수 있다.

Reference

1. Marc H. Lang, Peter E. McCormick, "Hierarchical Design Methodologies : A VLSI Necessity", Design Methodologies, Elsevier Science Publishers B.V. (North-Holland), 1986
2. C. Niessen, "Abstraction Requirements In Hierarchical Design Methodologies.", Design Methodologies, Elsevier Science Publishers B.V. (North-Holland), 1986
3. Jeffrey D. Ullman, "Computational Aspect of VLSI", Chap. 9, Computer Science Press.
4. "IBM Personal Computer Graphics Kernel System", IBM Personal Computer Software.

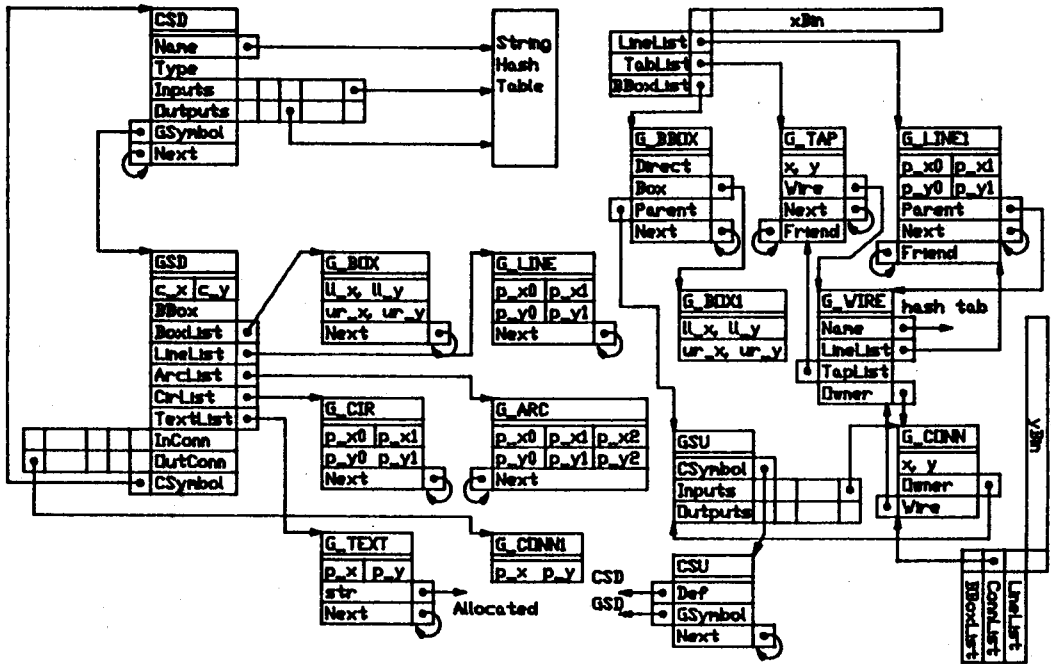


그림 1. Data 구조

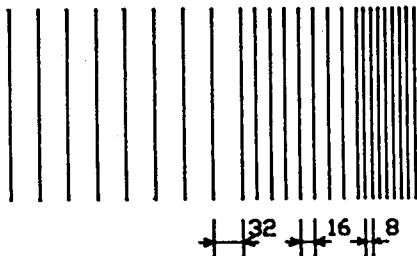


그림 2. bin 형태