

論文93-30A-1-9

# IC 설계용 집적형 카드 시스템의 구현

## (An Implementation of integrated CAD system for IC design)

孔鎮興\*, 金成重\*, 金載協\*

(Jin Hyeung Kong, Seong Jung Kim, and Jae Hyup Kim)

### 要約

본 논문에서는 집적회로 설계를 위한 카드 툴과 설계 환경을 집적시킨 카드 시스템의 설계 및 구현에 대하여 보인다. 카드 툴과 설계 환경은 공유 도메인(public-domain)에서 확보 가능한 것으로 하고 컴퓨터 시스템으로는 Unix와 X 윈도우 시스템을 탑재한 PC-386과 워크스테이션을 이용하여, 설계 자원이 제한된 환경(특히 교육기관)에 대한 카드 시스템의 보급 가능성을 높였다. 시스템의 카드 툴과 설계 환경은 오브젝트 구조와 변수 테이블을 이용하여 지정된 기능을 위한 지원 루틴과 데이터 파일을 프로그램함으로써 집적 및 편집이 가능하다. 설계 작업중 카드 툴 지원을 효과적(중분식)으로 얻을 수 있도록 툴 오브젝트 구조에는 툴간의 통신과 원형 순차 수행이 고려되었다. 현재 카드 시스템을 이용하여 LPC(Linear Predictive coding) 음성 합성기 집적회로 설계가 진행 중이며 시스템에 대한 개선 및 디버그가 추진되고 있다.

### Abstract

This paper presents a design and implementation of CAD(Computer-Aided Design) system with tools and design environments for IC(Integrated Circuits) design. The CAD system can be easily installed in various sites with limited resources, since most CAD tools and design environments are available in the public-domain and Unix & X Window-based PC-386 and Workstation is used for the hardware platform. In order to improve the flexibility of the CAD system, objects are defined in the context of tools and environments; and object tables are programmed to describe the integration of CAD tools and design environments. During the execution, tool-objects deal with intertool communication and round-robin mechanism to incrementally control the execution of CAD tools. The IC design of LPC(Linear Predictive Coding) Speech Synthesizer is carried out to find out improvements and bugs of the CAD system.

### 1. 서론

\*正會員, 光云大學校 컴퓨터 工學科  
(Dept. of computer Eng., wangwoon Univ.)  
接受日字: 1992年5月20日  
(∴ 본연구는 1990, 1991년도 교육부 반도체 분야  
학술연구조성비에 의하여 지원되었음.)

집적회로 설계 작업의 유효적이고 효과적인 수행이 가능토록 카드 툴들은 하나의 시스템을 구성하게 된다. 초기에 이 시스템은 특정 집적회로의 단위 설계 작업을 지원하기 위한 툴들의 임의(ad hoc) 집합 형태를 갖었다. 설계 과정의 유효적 수행을 위하여는

카드 톨간의 직접적인 호환성이 요구되었으나, 이 시스템은 호스트 컴퓨터의 운영체제를 통한 간접적인 호환성을 유지할 수 밖에 없기 때문에 집적회로 설계 작업에 대한 지원이 제한되었다. 카드 톨 간의 호환성을 위해 데이터베이스를 구조적으로 결합시킨 시스템<sup>13)</sup>이 제안되었다. 이러한 시스템은 특정 분야의 집적회로 설계 작업을 위한 전문가 시스템(expert system)<sup>2)</sup>으로 효과적이었으나, 상호간에 구조적으로 결합된 카드 톨이 시스템 개선에 대한 배타성을 수반하여 카드 시스템의 적응성을 제한했다. 카드 톨 간의 호환성과 시스템의 적응성을 추구한 카드 시스템이 1980년대 후반에 제안되었으며 카드 프레임워크(CAD framework)<sup>14)</sup>라 불린다. 이 시스템은 이질적이며 독자적인 카드 톨들을 기능적으로 결합시키고자 운영체제의 셸과 유사한 형태를 갖고 있으며, 복잡한 초대형 집적회로 설계 과정에서 요구되는 다양한 최적의 지원이 가능하도록 시스템의 카드 톨 및 설계 데이터베이스 관리를 설계자가 용이하게 직접 프로그래밍(운영) 할 수 있도록 했다.

본 논문에서는 집적회로 설계를 위한 카드 톨과 설계 환경을 집적시킨 중소형 카드 시스템을 제안하고 구현에 대하여 보인다. 카드 시스템이 높은 호환성과 이식성을 갖도록 Unix 운영체제와 X 윈도우 및 Motif 그래픽 시스템등 실질적인(de facto) 표준 시스템 소프트웨어를 개발 환경으로 했다. 컴퓨터 시스템으로는 중소형 컴퓨터 PC-386과 워크스테이션을 대상으로 하여 비교적 재원이 제한된 상황(특히 교육 기관)에서 집적회로 설계 교육 및 실습용 시스템이 보급될 수 있도록 하였다. 카드 시스템에서 설계 지원 기능을 수행하는 카드 톨과 설계 환경은 대부분 공유 도메인(public-domain)에서 확보 가능한 것으로 하였다. 또한 카드 톨과 설계 환경을 시스템에 탑재하기 위하여 오브젝트 구조를 이용했다. CAD-WELD 프레임워크 시스템<sup>15)</sup>에서 톨 집적을 위해 계층 톨 오브젝트 모델을 세안했는데, 이러한 톨 오브젝트 구조는 설계 지원을 위해 카드 톨들의 기능적 집적 및 수행 제어의 자동화를 도모한다. 이 모델은 설계자원이 방대한 시스템 구성을 위한 이질적인 설계 자원의 지원에는 효과적일 수 있으나 제한된 환경에 대하여는 시스템의 규모와 오버헤드가 과도하기 때문에 비효율적이기 쉽다. 본 논문에서는 중소형 컴퓨터 시스템에 다양한 카드 톨 및 설계 환경을 구조적으로 집적시키고 효과적인 수행제어(즉, 운영체제의 라운드 로빈 방식)를 가능토록 하기 위한 단층 오브젝트 구조를 집적회로 설계 환경과 카드 톨에 대하

여 제안하고 구현하였다.

카드 시스템은 집적회로 설계를 지원하기 위해 다음의 주요 특성을 구현하였다.

- 오브젝트 지향적(object-oriented) 구성
- 다양한 집적회로 설계 환경 집적
- 다양한 카드 톨 집적 및 효과적 수행 제어
- 중앙 집중적 관계형 데이터베이스의 구조적 공유
- 중앙 집중적 GUI(Graphic User Interface) 구조

본 논문에서 제 II 장은 시스템 개요, 제 III 장은 카드 시스템 구성에 대하여 논하였다. IV 장에서는 구현된 카드 시스템의 특성 평가 및 기존 카드 시스템에 비하여 평가하고, 시스템을 이용한 집적회로 설계 예를 보인다. 마지막으로 시스템에 대한 고찰과 보완점에 대하여 생각한다.

## II. 시스템 개요

본 시스템은 이식성을 유지 하기 위하여 실질적인 표준 운영체제와 그래픽 시스템을 소프트웨어 환경 시스템으로 이용하였다. 운영체제(OS: Operating System)로는 Unix 계열의 System V/386과 BSD 4.3과 같은 중소형 컴퓨터의 운영체제를 사용하였으며, 그래픽 시스템 환경으로는 X11.3 윈도우 시스템과 Motif 1.0을 이용하였다. 카드 시스템을 위한 컴퓨터 소프트웨어 환경 시스템의 구성은 그림 1(a)와 같다. 현재 시스템은 IBM-PC/386과 Apollo 워크스테이션에 탑재되어 있다. 카드 시스템은 C언어를 사용하여 그림 1(b)와 같은 3 층의 셸(shell) 구조로 구현되어 있다. 시스템의 중심부는 설계 환경 데이터와 실제 설계 데이터로 구성된 공통 데이터베이스이며, 두번째 셸에서는 시스템의 데이터베이스를 관리하는 루틴들이 카드 톨과 데이터베이스 간의 상호 동작을 지원 처리한다. 확보된 기존 카드 톨과 시스템 고유의 지원을 위해 구현된 톨로 외곽 셸을 구성한다. 시스템의 설계 환경으로 Mead & Conway의 nMOS<sup>16)</sup>와 Griswold의 n/p-well CMOS<sup>18)</sup> 및 Mosis 2층 메탈 p-well CMOS<sup>19)</sup>가 집적회로 레이어아웃 설계를 지원하며 디지털 로직 및 아날로그 회로 구성을 위한 스키매틱 환경<sup>1)</sup>이 포함되어 있다. 집적회로 설계 작업을 지원하기 위하여 사용자 톨과 네트워크 톨, DRC(design rule checker), 압축기, 시뮬레이터 톨 및 입출력 톨이 카드 시스템에 집적되어 있다. 사용자 톨은 시스템 운영과 집적회로 편집 및 카드 톨 수행제어 기능을 담당하는 GUI이며, 네트

워크 툴은 설계된 회로의 연결 정보를 유지 관리하여 다른 툴들에게 제공하는 역할을 수행한다. DRC 툴<sup>11</sup>은 집적회로 레이아웃 데이터의 설계 툴을 점검하며, 압축기<sup>12</sup>는 레이아웃 면적을 1차원(X축/Y축) 방향으로 최적화 한다. 시뮬레이터 툴은 기존 회로 시뮬레이터들의 수행을 제어하고 입력 파일을 발생하는 역할을 수행한다. 현재 ESIM<sup>10</sup>, RSIM<sup>111</sup>, MOSSIM<sup>12</sup>, COSMOS<sup>13</sup>와 SPICE<sup>14</sup> 시뮬레이터가 시스템에 탑재될 수 있다. 외부에 대한 설계 데이터의 호환성을 유지하기 위해 표준 데이터 형식인 CIF(Caltech Intermediate Format)<sup>115</sup>와 EDIF(Electronic Design Intermediate Format)<sup>115</sup>의 파싱 및 변환 기능을 입출력 툴이 수행한다.

CAD system.

### Ⅲ. 캐드 시스템 구성

이 장에서는 캐드 시스템의 구성을 데이터베이스와 설계 지원 캐드 툴로 나누어서 논한다. 데이터베이스 절에서는 집적회로 표현을 위한 데이터 구조와 설계 환경의 오브젝트에 대하여 설명하고, 캐드 툴 절에서는 툴의 집적과 효과적 수행 제어를 위한 툴 오브젝트 구조와 인터페이스 메카니즘에 대하여 논하고 시스템의 설계 자동화 범위를 밝히기 위해 탑재된 툴에 대하여 설명하고 있다.

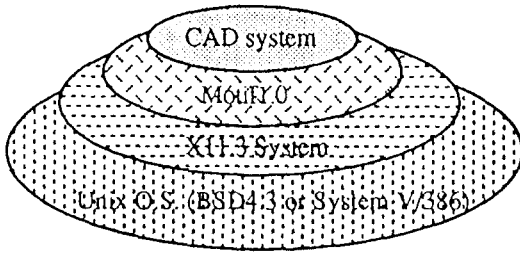
#### 1. 데이터베이스

캐드 시스템의 데이터베이스는 집적회로 설계 데이터를 기술하는 부분과 시스템의 설계 환경을 표현하는 부분으로 구성된다.(그림 1(b)) 설계 데이터 및 환경은 오브젝트 지향적 구조를 갖고 오브젝트들 간의 상호 관계를 연결 리스트 구조로 설정하는 관계형 데이터베이스 형태로 구현된다.

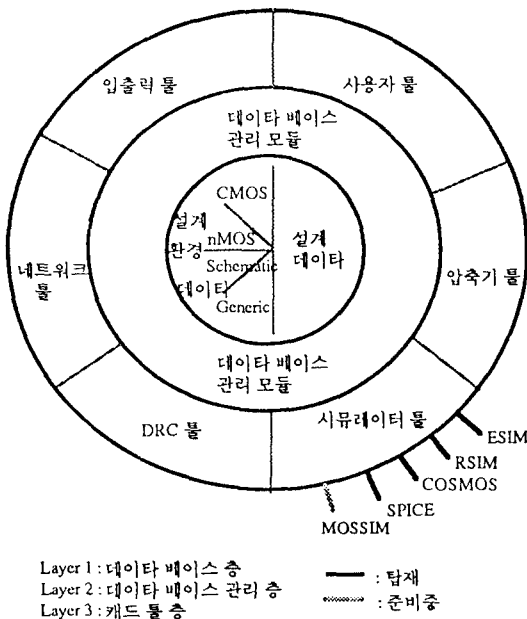
#### 1) 데이터 구조 및 처리 루틴

시스템은 집적회로 구조 특성인 계층성과 반복성 및 모듈성을 설계 작업에서 효과적으로 처리할 수 있도록 라이브러리와 셀, 피셋, 원시 단위 및 테크노로지의 5레벨 개념 모델을 이용한다. 가장 상위 계층의 라이브러리는 전기적 회로 단위에 해당하는 셀들로 구성되며 하나의 셀은 여러가지 피셋을 포함하게 되는 데 각 피셋은 서로 다른 견지(예: 레이아웃, 스키메틱, 아이콘 등)로 회로의 실체를 기술하게 된다. 피셋은 설계 환경의 원시 단위로 구성되며, 원시 단위들은 테크노로지에 의하여 모든 특성이 표현된다. 테크노로지는 설계 환경을 조성하기 위한 정보를 갖고 있는데 설계 룰 테이블(design-rule table)과 기하 형태 기술(geometric description)로 구성된다.

캐드 시스템에서 회로는 노드와 아크로 구성된 네트워크로 표현된다.(그림 2) 노드는 트랜지스터나 로직 게이트 및 콘택 등과 같은 전기적 단위에 해당되며, 아크는 노드에 선언되는 포트를 통하여 노드간을 연결하는 선을 의미한다. 노드는 프로토타이프(prototype) 오브젝트와 인스턴스(instance) 오브젝트로 처리되는 데, 프로토타이프 오브젝트는 테크노로지에 의하여 기술되는 설계 환경의 원시 단위 노드에 해당하며 인스턴스 오브젝트는 프로토타이프 오브젝트와 다른 인스턴스 오브젝트로 구성된다. 아크는



(a) 캐드 시스템의 소프트웨어 환경구축



(b) 캐드시스템의 구조

그림 1. 캐드시스템의 환경과 구조

Fig. 1. Environments and Architecture of

프로토타이프 오브젝트로 처리되어 프로토타이프나 인스턴스 오브젝트 노드간의 연결을 표현하게 된다. 시스템에서 설계 과정은 데이터베이스에 대한 오브젝트 구성과 포인터 지정 작업을 통하여 수행된다. 회로 구성을 위한 오브젝트는 프리 메모리로 부터 할당 되고 오브젝트 간의 상호 관계는 단방향 혹은 쌍방향 연결 리스트 구조로 지정한다. 설계된 회로의 크기가 제약받지 않도록 고정된 배열 구조는 이용하지 않았다. 카드 시스템에서 집적회로는 회로 구조 및 동작 정보를 갖은 오브젝트들을 연결한 구조에 해당한다. 또한 회로 구성을 그래픽 형태로 처리 할 수 있도록 기하 정보를 갖은 오브젝트를 이용한다. 설계 데이터의 구성을 위한 기술 오브젝트들은 그림 3의 3가지 종류로 분류된다.

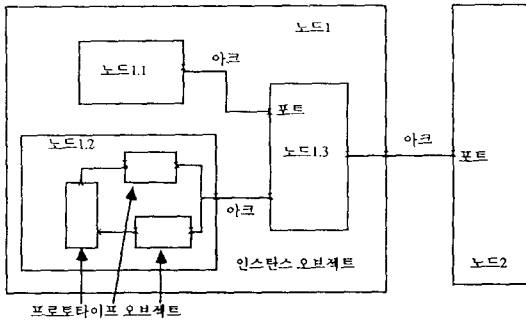


그림 2. 카드 시스템의 회로 구성 예  
Fig. 2. A representation of IC circuit in CAD system.

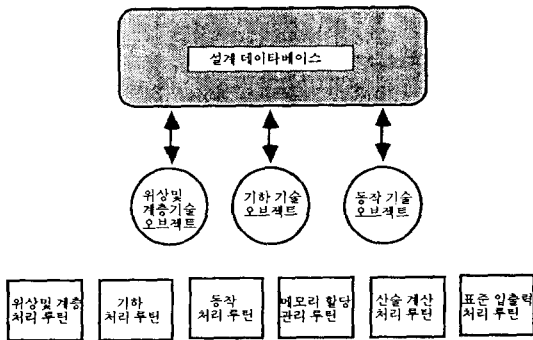


그림 3. 데이터베이스의 오브젝트와 처리루틴  
Fig. 3. Objects and Routines in Design Database.

.위상 및 계층 기술 오브젝트들  
노드와 포트 및 아크의 프로토타이프 및 인스턴스

오브젝트와 셀 및 라이브러리 오브젝트, 노드간 연결을 표현하는 네트워크 오브젝트등으로 구성되어 집적회로의 위상 (topology) 연결 구조와 계층 구조를 기술한다.

.기하 기술 오브젝트들

노드와 아크 인스턴스 오브젝트들의 위치와 경계를 기술하는 기하 모듈 오브젝트, 기하 모듈을 효과적으로 저장 및 검색하기 위한 균형 트리(balanced tree) 오브젝트<sup>[16]</sup>, 오브젝트를 그래픽으로 기술하는 폴리곤 오브젝트 등으로 구성된다. 이 오브젝트들은 설계 데이터의 기하 정보를 처리(저장, 검색)하기 위하여 이용된다.

.동작 기술 오브젝트들

셀 퍼셋의 실체(즉 레이아웃, 스키메틱, 아이콘)를 나타내는 견지 오브젝트, 설계 환경의 원시 단위들을 기술하는 테크노로지 오브젝트, 각 오브젝트의 속성 및 특성으로 구성된 변수 오브젝트 및 데이터베이스의 모든 변화를 계층성에 따라 처리하기 위한 데몬 오브젝트로 이루어 진다. 이 데이터 구조들은 회로 설계 데이터를 구성하는 오브젝트의 실체, 환경과 속성 정보를 표현 및 처리한다.

데이터베이스의 오브젝트들을 처리하기 위한 루틴들은 모듈화되어 구현되어 있다.(그림 3) 위상 및 계층 오브젝트와 기하 오브젝트 및 동작 오브젝트에 대한 조작과 정보 입력 및 검색 등을 처리하는 루틴, 행렬 데이터의 조작과 일반적인 수학 계산을 지원하는 루틴, 오브젝트에 대한 메모리 할당을 시스템 내의 각 오브젝트들의 메모리 사용 위치 변수를 이용하여 관리하는 루틴, 컴퓨터 시스템의 디스크 파일을 쓰고 읽는 과정을 수행하는 표준 입출력 루틴 등으로 구성되어 있다.

2) 설계 환경

현재 시스템은 레이아웃 설계를 위한 Mead /Conway의 nMOS<sup>[7]</sup>와 Griswold의 n/p-well CMOS<sup>[8]</sup> 및 MOSIS 2층 메탈 p-well CMOS 설계 환경<sup>[9]</sup>과 디지털 로직 및 아날로그 전자회로 구성을 위한 스키메틱<sup>[11]</sup> 설계 환경을 제공한다. 각 설계 환경은 데이터베이스의 동작 기술 오브젝트중 하나인 테크노로지 오브젝트(그림 4(a)) 구조로 구현된다. 테크노로지 오브젝트는 설계 환경 고유의 특성 값과 원시 단위 구조 정보 및 지원 루틴등을 기술하고 있다. 카드 시스템에서 지원되는 설계 환경(nMOS, CMOS, Schematic 등)들에 대한 테크노로지 오브젝트의 값들은 설계 환경 변수 테이블(eltech-

nologylist)에 등록되어 있다.(그림 4(b)) 오브젝트의 변수를 통해 검색되는 설계 환경 기술 데이터는 목차형(tabular) 정보와 환경 지원 루틴으로 구성된다. 목차형 정보는 설계 환경 레이어의 패턴 및 칼라(layer), 설계 룰(tchdescript), 원시 노드와 아크 구조(~protos) 등의 데이터 파일로 구현되어 있고 지원 루틴(init)은 목차형 정보를 이용하여 카드 시스템에 설계 환경을 조성한다.(그림 4(a))

```
typedef struct technology{
  char *techname; /* name of this technology */
  INT16 index; /* 0 based index of this technology */
  INT32 deflambd; /* the default size of a unit */
  NODEPROTO *firstnodeproto; /* list of primitive nodeprotos */
  ARCPROTO *firstarcproto; /* pointer to type description */
  VARIABLE *firstvar; /* first variable in list */
  INT16 numvar; /* number of variables in list */
  COMCOMP *parse; /* parsing structure for tech direction */
  CLUSTER *cluster; /* virtual memory cluster for technology */
  char *techdescript; /* description of this technology */

  INT16 layercount; /* number of layers */
  GRAPHICS **layers; /* layer descriptions */
  INT16 arcprotocount; /* number of arcs */
  TECH_ARCS **arcprotos; /* raw data for arcs */
  INT16 nodeprotocount; /* number of nodes */
  TECH_NODES **nodeprotos; /* raw data for nodes */
  TECH_VARIABLES *variables; /* variable descriptions */

  INT16 (*init)(); /* process initialization */
  void (*setmode)(); /* set operation mode */
} TECHNOLOGY;
```

(a) 테크노로지 오브젝트

```
TECHNOLOGY el technologylist[] =
{
  /* NMOS */
  "nmos", 0, 200,
  NONODEPROTO, NOARCPROTO, NOVARIABLE, 0, NOCOMCOMP, NOCLUSTER,
  "n-channel MOS (from Mead & Conway)",
  15, nmos_layers, 3, nmos_arcprotos, 26, nmos_nodeprotos, nmus_variables, 0, 0,

  /* CMOS */
  ...

  /* MOSIS CMOS */
  "Round MOSIS CMOS",
  ...
  /* Schematic capture */
  ...
};
```

(b) 설계 환경 변수 테이블

그림 4. 설계환경 오브젝트와 변수 테이블  
Fig. 4. Technology Object and Variable table.

시스템의 모든 설계 환경은 공통 구조의 테크노로지 오브젝트로 표현되며 오브젝트의 변수 값에 따라 각 설계 환경을 구현하게 된다. 이러한 오브젝트와 변수 테이블 구조에 의한 설계 환경의 집적은 편집의 가능성을 보인다. 설계 환경의 목차형 정보를 편집하고 변수 테이블에 새로운 환경 오브젝트의 변수 데이

타를 등록하면 시스템의 설계 환경의 확장 및 변경이 가능하다. 카드 시스템에 대한 설계 환경의 목차형 정보 편집을 처리 할 수 있는 전용 편집기의 개발을 진행하고 있다.

2. 카드 툴

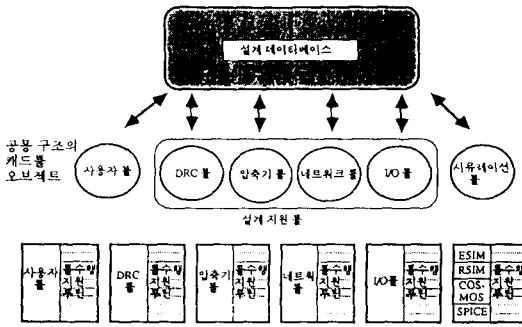
카드 시스템은 설계자의 집적회로 설계 작업을 직접 및 간접으로 지원하는 다양한 카드 툴(그림 5)들을 탑재하고 있다. 카드 툴은 사용자 툴과 설계 지원 툴 및 시뮬레이터 툴로 구분된다.

1) 카드 툴 집적 및 인터페이스

카드 툴들은 공유된 데이터베이스에 대하여 설계 데이터의 효과적 처리(편집, 검증)를 할 수 있도록 시스템에 구조적으로 집적되었다. 시스템이 지원하는 카드 툴은 공통 구조(그림 6(a))의 오브젝트 형태로 구현되며, 오브젝트의 변수는 카드 툴과 툴 수행 지원 루틴 및 툴의 상태등을 지정하고 있다. 시스템 카드 툴들에 대한 오브젝트 변수 값은 카드 툴 변수 테이블(el\_aids)에 등록되어(그림 6(b)) 각 카드 툴의 고유 오브젝트 구성(그림 5)이 가능하다. 각 툴 오브젝트 변수에 지정된 카드 툴과 툴 수행 지원 루틴은 카드 툴의 동작을 제어하게 되는 데, 카드 툴 수행(slice)은 집적회로 설계 작업을 직접적으로 지원하는 특유의 기능을 수행시키며 툴 수행 지원 루틴들은 시스템과 카드 툴간의 인터페이스 동작을 처리한다.(그림 6(a)) 툴 오브젝트의 동작은 툴 수행 조건(aidstate), 툴 초기화 및 완료루틴(init/done)과 데이터베이스 변경 및 기록 루틴(~change, ~batch), 설계 데이터 변경 전달 루틴(/"Intercommunication routines"/)등에 의하여 수행된다.(그림 6(a))

카드 프레임워크<sup>[4]</sup>에서 툴 수행의 기능적 제어를 자동화하기 위해 계층 툴 오브젝트 모델을 이용하고 있으나, 본 시스템에서는 그림 6의 툴 오브젝트를 통하여 구조적으로 결합된 툴의 편집 및 제어 메카니즘을 효과적으로 처리한다. 툴 오브젝트의 이용은 시스템의 카드 툴에 대한 편집 가능성을 높인다. 툴과 시스템간의 인터페이스 동작을 위한 루틴의 지정된 역할에 맞추어 툴 수행 지원 루틴들을 구현하고 루틴들을 카드 툴 변수 테이블에 등록함으로써 임의의 카드 툴을 시스템에 탑재할 수 있다. 또한 공통 오브젝트 구조를 각각의 툴에 대하여 특성화시켰기 때문에 시스템에 탑재된 다양한 툴들에 같은 제어 방법을 적용할 수 있도록 한다.

카드 툴에 대한 인터페이스는 그림 7과 같은 공통 구조를 갖고 있다. 사용자는 카드 시스템에 탑재되어 있는 툴의 오브젝트(그림 6(a))에 대하여 인터페이스 제어 스위치(aidstate)를 ON/OFF 할 수 있다. 스위치가 ON되어 있는 툴만을 인터페이스에서 제어하게 된다. 인터페이스는 운영체제의 라운드 로빈 방식(그림 7(b))을 이용하여 ON 카드 툴을 원형 순차적으로 수행(slice)시킨다. 각 툴은 수행 기회가 주어질 때 데이터베이스 변경 루틴을 통해 설계 데이터를 수정 및 편집 하고 데이터베이스의 변경 전달 루틴은 변동 사항을 다른 툴에 알려 준다. 데이터의 변동 사항을 전달받은 툴은 수행 기회가 주어질 때 까지 기다린다. 카드 시스템에 탑재된 모든 툴에 대한 공통 인터페이스 동작 메카니즘은 다음과 같다.



카드 툴과 테이블에 등록된 수행지원루틴

그림 5. 카드 툴 오브젝트와 처리루틴  
Fig. 5. CAD Tool Objects and Routines

- step 1 : 모든 카드 툴의 오브젝트를 초기화한다.  
이때 각 툴의 오브젝트(그림 6(a))에 변수 테이블에 등록된 데이터(그림 6(b))를 입력한다.
- step 2 : 원형 순차 기회를 각 툴에 부여하는 동작을 반복한다.  
이 과정은 시스템이 동작하는 동안 무한 루프 수행을 한다.
- step 2.1 : 선택된 카드 툴의 오브젝트(el\_aids [ ] )를 찾는다.
- step 2.2 : 카드 툴의 제어 스위치(aidstate)가 ON 인지를 검사한다.  
OFF되어 있으면 step 2.8로 간다.
- step 2.3 : 카드 툴이 수행(slice)된다.  
이때 데이터베이스를 검색 및 변경할 수

```
typedef struct{
char *aidname; /* name of aid */
INT32 aidstate; /* state of aid */
INT16 index; /* aid index */
COMCOMP *parse; /* parsing structure for aid direction */
CLUSTER *cluster; /* virtual memory cluster for this aid */

void (*init)(); /* initialization */
void (*done)(); /* completion */
INT16 (*scunode)(); /* user-command options */
INT32 (*request)(); /* direct call options */
void (*examinenodeproto)(); /* to examine an entire facet at once */
void (*slice)(); /* time slice */

void (*startbatch)(); /* start change broadcast */
void (*endbatch)(); /* end change broadcast */
/* Intercommunication routines */
...
} AIDENTRY;
```

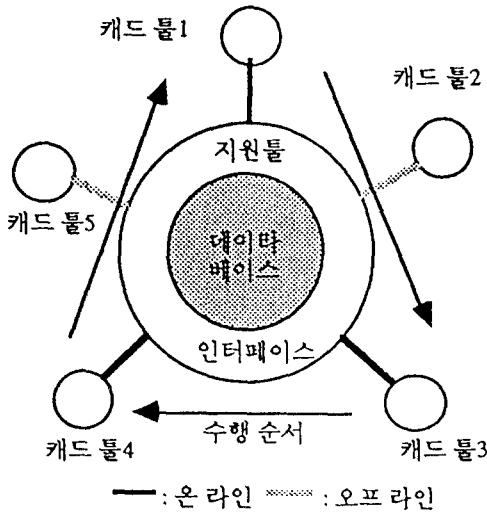
(a) 카드 툴 오브젝트

```
AIDENTRY el_aids[] =
{
/* user interface */
"user", AIDON | AIDINCREMENTAL, 0, &us_userp, NOCLUSTER,
us_init, us_done, us_sel,
us_request, us_examinenodeproto, us_slice,
us_startbatch, us_endbatch,
...
/* I/O module */
...
/* compaction module */
...
/* network maintenance and comparison */
...
/* design-rule checker */
...
/* simulation module */
...
};
```

(b) 카드 툴 변수 테이블

그림 6. 카드 툴 오브젝트와 변수테이블  
Fig. 6. CAD Tool Object and Variable Table.

- 있다.
  - step 2.4 : 툴의 수행이 설계 데이터를 변경했는가 (batch)를 검사한다.
  - step 2.5 : 변경 사항이 없으면 step 2.8로 간다.
  - step 2.6 : 변경 사항이 있으면 설계 데이터의 계층 구조를 통한 부수적 변경 사항을 감지하여 데이터베이스를 변경한다. (change)
  - step 2.7 : 데이터베이스를 변경시킨 툴의 수행 내용을 ON되어 있는 모든 카드 툴에 전달 (broadcast)한다.
  - step 2.8 : 원형 순차 수행 순서에서 다음 툴을 선택 될 카드 툴로 지정하고 step 2.1로 간다.
- 이러한 카드 툴 인터페이스 방식은 다음과 같은 주



(a) 카드 툴 인터페이스

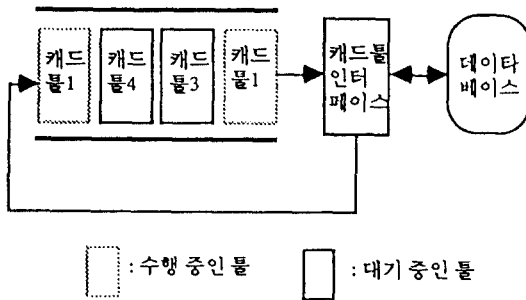


그림 7. 카드 툴 인터페이스 구조 및 제어방식

Fig. 7. CAD Tool interface and control mechanism.

요 특성을 보인다.

- ① 시스템에 집적된 카드 툴이 증분식 수행 (incremental execution)을 한다 :  
수행 기회(slice)가 원형 순차적으로 부여된다.
- ② ON 카드 툴간에 작업 내용을 상호 교환한다 :  
수행 기회가 부여된 툴의 작업 내용은 전달 루틴 (broadcast)에 의하여 다른 카드 툴에 알려진다.
- ③ 설계 작업의 과정 중에 다양한 카드 툴의 지원을 동시에 받을 수 있다 :  
예를 들어 사용자 툴이 집적회로 레이아웃 구성을 변경한 경우에 ON DRC 툴은 변경된 데이터가 설

계 룰에 맞는지를 검사하고 네트워크 툴은 회로의 연결이 변경된 사항을 검색하여 사용자 툴이 수행 될 때 설계 룰 검사 결과와 연결 사항을 설계자에게 알려 준다.

2) 사용자 툴

카드 시스템은 그래픽 디스플레이와 키보드 및 마우스로 구성된 하드웨어 인터페이스에 대하여 구현된 사용자 툴 GUI(Graphic User Interface)를 지원한다. 사용자 툴의 디스플레이는 상태 윈도우와 편집 윈도우로 구성된다. 텍스트 입출력을 위한 상태 윈도우는 두 영역으로 나뉘어 지는데, 설계 환경의 사상을 보여 주는 고정 영역과 작업을 지시하는 명령어 입력 및 시스템의 상태 정보를 출력하는 스크롤 영역으로 구성된다. 편집 윈도우는 칼라/흑백 그래픽 입출력을 처리하게 되는데 메뉴 영역과 작업 영역으로 나누어 진다. 메뉴 영역은 집적회로 설계 환경의 원시 단위와 그래픽 명령어의 아이콘으로 구성되며, 작업 영역은 회로의 구조를 보여주고 편집이 수행되는 부분이다. 사용자 툴의 데이터 구조(그림 8)는 기본적인 디스플레이 데이터 구조인 윈도우 오브젝트와 집적회로 설계 데이터를 그래픽 입출력으로 처리하기 위한 그래픽 오브젝트 및 카드 시스템의 명령어를 해석하기 위한 오브젝트로 구성되며, 오브젝트들은 상태 윈도우를 통한 텍스트 정보의 입출력과 키보드 입력을 위한 터미널 입출력 루틴과 마우스 입력을 처리하는 루틴 및 여러가지 텍스트 스트링 처리를 위한 루틴등에 의하여 다루어 진다.

3) 설계 지원 툴

카드 시스템의 설계 지원 툴은 네트워크 툴과 DRC, 압축기 및 입출력기로 구성된다. 설계 지원 툴들은 데이터베이스를 검색하거나 변경하는 기능을 수행할 수 있도록 시스템에 집적되어 있다.

네트워크 툴은 위상 및 계층 오브젝트 중의 하나인 네트워크 오브젝트를 관리하고 시스템의 카드 툴에게 설계 데이터에 대한 연결 정보를 제공한다. 설계 데이터의 연결 정보를 관리하기 위해 툴 인터페이스의 증분식(incremental) 제어에 따라 오브젝트의 아크와 포트 변경 사항을 주기적으로 검사한다. 시스템의 카드 툴(사용자 툴, DRC, 압축기, 시뮬레이터 툴)은 네트워크 툴에게 설계 데이터의 연결 정보를 요구하거나 데이터베이스의 변경 사항을 알려 준다. 레이아웃 설계 데이터의 기하 정보가 특정 집적회로 설계 환경의 설계 룰을 위반하였는지를 검사하는 기능을

DRC 툴<sup>[1]</sup>이 담당한다. 이 툴은 설계자가 사용자 툴을 이용하여 설계 데이터를 편집할 때 증분식으로 수행하면서 오류를 검사하여 준다. 집적회로 레이아웃 면적을 최적화하기 위하여 카드 시스템은 설계 데이터를 X축 및 Y축으로 압축시키는 1차원 압축기<sup>[11]</sup>를 포함하고 있다. 입출력기는 카드 시스템의 설계 데이터와 호환성이 우수한 CIF 및 EDIF 데이터 간의 변환 기능을 담당하며, 각 형식에 대한 파싱과 입력 및 출력 기능을 수행한다.

4) 시뮬레이터 툴

이 툴은 기존의 시뮬레이터들을 카드 시스템에 탑재하기 위하여 구현되었다. 시뮬레이터 툴은 다른 툴과는 달리 데이터베이스를 변경할 수 없고, 데이터베이스를 검색하며 탑재된 기존 시뮬레이터의 입력 파일(즉 넷리스트)을 발생하는 기능을 수행한다. 현재 넷리스트 발생이 ESIM<sup>[10]</sup>, RSIM<sup>[11]</sup>, MOSSIM<sup>[12]</sup>, COSMOS<sup>[13]</sup>, 및 SPICE<sup>[14]</sup>에 대하여 가능하며, MOSSIM을 제외한 시뮬레이터들이 카드 시스템에 탑재되어 있다. 시뮬레이터의 수행은 카드 시스템 프로세스가 시뮬레이터 프로세스를 발생(fork)시켜 이루어 진다. 카드 시스템에 다른 시뮬레이터를 탑재하기 위해서는 시뮬레이터 툴에 집적된 시뮬레이터로 등록하고 전용의 넷리스트 발생 루틴을 시스템에 추가하여야 한다. 이러한 종속 프로세스 인터페이스 방식은 기존 시뮬레이터의 탑재와 제어를

어 있는데, 현재 SPICE->RSIM 변환기가 시스템에 포함되어 있다. 이 변환기는 다른 카드 시스템에서 발생한 SPICE 넷리스트에 대하여 RSIM 시뮬레이션이 가능하게 한다.

IV. 시스템 특성 평가

1) 카드 시스템 특성

본 논문에서 설계된 카드 시스템은 다음과 같은 주요 특성을 갖고 있다.

- ① 시스템은 관계형 데이터베이스를 포함한다.

관계형 데이터베이스는 집적회로의 구조 특성인 계층성과 반복성, 모듈성 및 다양한 개념 모델을 처리할 수 있다. 본 시스템은 5레벨의 개념 모델(라이브러리, 셀, 퍼셋, 원시 단위, 설계 환경)을 바탕으로 집적회로의 계층성을 표현하며, 하나의 개념 모델을 회로 내에 반복적으로 인스턴스하고 포트의 외부 연결을 통해 기능을 프로그램할 수 있어 집적회로의 반복성 및 모듈성을 효과적으로 처리할 수 있다. 또한 시스템에서 제공 하는 다양한 설계 환경(예:스키메틱(스위치 및 게이트), 레이아웃(nMOS, CMOS))을 이용하여 집적회로를 다양한 견지에서 표현할 수 있다.

관계형 데이터베이스는 집적회로 구조 정보의 효과적 검색과 관리 및 유지를 용이하게 한다. 예를 들어, 회로의 연결 정보를 유지 및 관리하는 네트워크 툴은 아크와 포트 오브젝트 만을 이용하여 설계 회로에 대한 연결 상태를 다른 카드 툴(사용자, DRC, 시뮬레이터등)에 제공할 수 있다. 또한 설계 데이터의 일부가 변경되었을 때 연결된 구조만을 추적하여 상위 계층 구조에 관련된 변경 사항을 처리할 수 있어 계층 구조 설계 데이터의 일관성 유지가 용이하다. 이러한 데이터베이스의 특성은 대형 집적회로 설계를 계층 하향적(Top-down)으로 진행할 수 있도록 하여 준다.

- ② 시스템은 오브젝트 지향적 구성과 연결 리스트 구조를 이용한다.

카드 시스템은 모든 구성원(설계 데이터, 카드 툴 등)을 오브젝트로 구현 및 처리하고 단방향 혹은 쌍방향 연결 리스트 구조로 관련성을 지정한다. 집적회로 설계 데이터는 3 가지 기술 오브젝트로 표현된다. 위상 및 계층 오브젝트는 회로 구조를 기술하며, 기하 오브젝트는 회로의 그래픽 정보를 표현하고, 동작 오브젝트는 설계 환경 및 데이터의 특성을 설명한다.

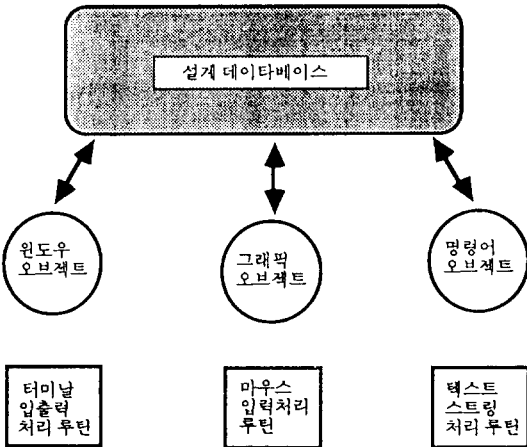


그림 8. 사용자 툴 오브젝트와 처리루틴

Fig. 8. User Tool Objects Routines.

용이하게 하며 시뮬레이터 고유의 동작 환경을 설계자가 이용할 수 있도록 하기 위함이다. 시뮬레이터 툴에는 시뮬레이터 넷리스트 간의 변환기가 탑재되



이러한 오브젝트들은 필요할 때 마다 실체가 구성되고 연결 리스트 구조로 서로 간의 관련성이 지정되면서 확장 가능하기 때문에 큰 집적회로를 카드 시스템에서 처리할 수 있다. 시스템에 탑재된 툴의 제어 및 수행을 위하여 카드 툴의 기능 및 시스템과의 상호 동작을 지정한 오브젝트를 카드 툴에 대하여 구성한다. 이외에도 그래픽 사용자 인터페이스(GUI)를 위한 윈도우 및 그래픽 오브젝트와 명령어를 파싱하기 위한 오브젝트가 이용된다.

③ 카드 시스템의 설계 지원 기능을 수행하는 카드 툴과 설계 환경의 집적은 오브젝트 구조를 이용한다.

카드 툴은 공통의 카드 툴 오브젝트 구조를 변수 테이블로 부터 특성화하여 고유의 오브젝트를 구성한다. 툴 오브젝트는 툴의 구조적 결합(데이터베이스와의 상호 작용과 작업 내용을 다른 툴에 전달하는 기능) 및 툴 수행 제어등의 정보를 갖는다. 카드 툴 인터페이스 프레임은 툴 오브젝트를 제어하는데, 오브젝트의 수행 제어 스위치를 ON/OFF로 프로그램할 수 있으며 ON 카드 툴에게 라운드 로빈 방식의 원형 순차 수행 기회를 준다. 수행된 툴은 툴 오브젝트를 통해 데이터베이스를 검색 및 변경하며 다른 툴에게 작업 내용을 알린다. 이러한 툴의 순차 수행과 툴 간의 상호 통신은 설계 작업 중에 여러 카드 툴들의 증분식 지원이 가능토록 한다.

시스템의 모든 설계 환경은 공통 구조의 환경 오브젝트 구조를 갖는다. 각 설계 환경은 오브젝트의 변수 값을 설계 환경 변수 테이블로 부터 지정하여 구현한다. 카드 시스템은 환경 오브젝트를 통해 집적회로 설계 환경을 기술하는 특성 값과 구조 정보(목적형 정보)를 검색하고 지원 루틴을 수행하여 설계 환경을 조성한다.

공통 오브젝트 구조와 변수 테이블을 통한 카드 툴과 설계 환경의 집적 방식은 카드 시스템에 대한 편집 가능성을 보인다. 오브젝트 변수의 특성 및 기능이 정의 되어 있기 때문에 해당되는 데이터와 지원 루틴을 구성하여 변수 테이블에 등록시킴으로써 카드 툴과 설계 환경을 편집할 수 있다.

④ 본 카드 시스템은 이식성을 유지하기 위하여 실질적인 표준 소프트웨어 개발 환경을 이용 하였다.

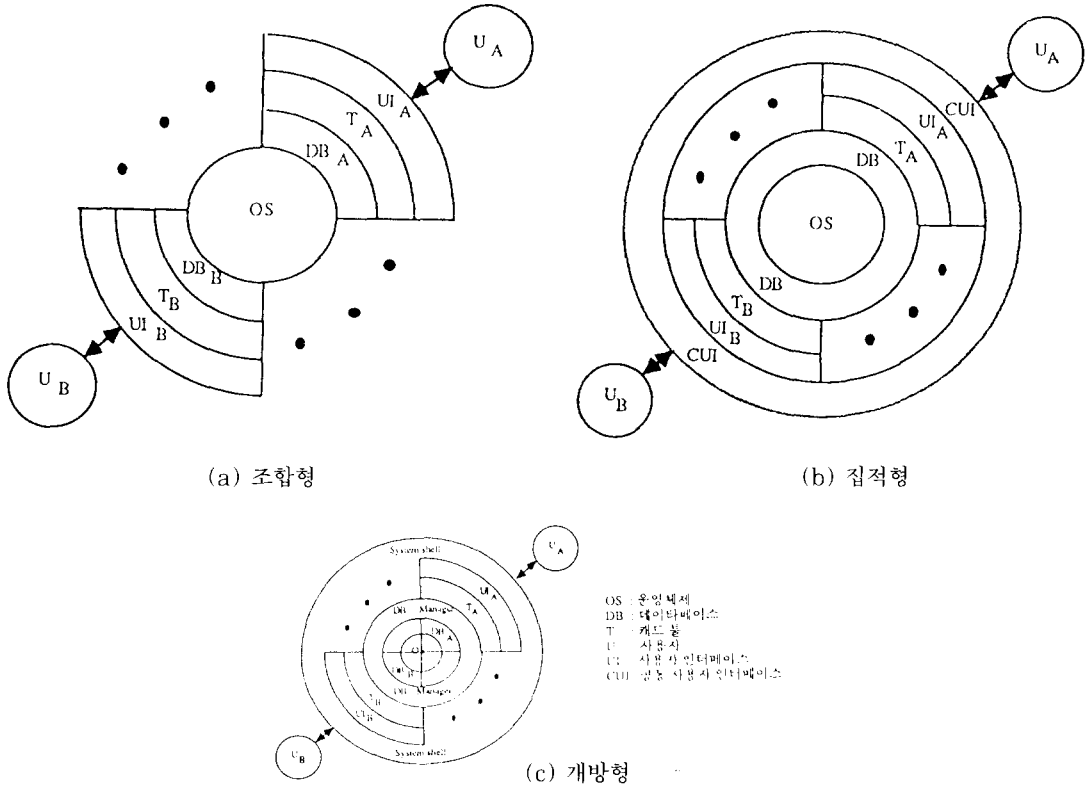
운영체제(OS: Operating System)로는 Unix 계열의 System V/386과 BSD 4.3과 같은 중소형 컴퓨터의 운영체제를 사용하여 카드 시스템의 호환성을 고려하였고, 그래픽 시스템으로는 별도의 그래픽 하드웨어를 요구하지 않는 X 윈도우 시스템과 Motif를 이용하여 시스템 GUI의 이식성을 유지하고자 했다.

현재 시스템은 IBM-PC/386과 Apollo 컴퓨터에 탑재되어 있는데, 컴퓨터 기종으로 PC-386과 같은 소형 보급 기종을 선택하여 시스템의 확대 보급을 도모하였다.

## 2) 시스템 평가

기존의 카드 시스템은 크게 3가지로 구분된다: (1) 조합형(Assembled) 카드 시스템, (2) 집적형(Integrated) 카드 시스템, (3) 카드 프레임워크(CAD framework). 조합형 카드 시스템은 카드 툴의 임의(ad hoc) 집합 형태(tool set)로서 운영체제를 바탕으로 지원 툴들이 시스템을 구성한다.(그림 9(a)) 이 시스템은 이질적인 툴들이 고유의 데이터 구조를 유지하기 때문에 툴 상호간의 호환성이 제한되어 설계 작업에 대하여 유기적인 지원을 제공할 수 없다. 설계 데이터의 호환성을 해결하기 위하여 집적형 카드 시스템<sup>1)</sup>은 데이터베이스를 공유하면서 카드 툴간의 구조적 결합을 시도하였다.(그림 9(b)) 시스템의 카드 툴 간의 구조적 결합은 시스템의 구성 툴에 대한 편집(확장, 제거등)을 제한하여 시스템의 활용 범위가 광범위한 경우에 한계를 갖게 되었다. 시스템의 적응성 문제를 해결하기 위하여 이질적인 설계 자원(카드 툴, 데이터베이스, 컴퓨터 시스템)을 기능적으로 결합(그림 9(c))시키려는 카드 프레임워크 시스템<sup>2)</sup>이 제안되었다. 프레임워크 시스템은 운영체제의 셸과 같은 역할을 수행하여, 이질적인 카드 툴을 제어하고 툴 고유의 데이터베이스를 관리하며 설계 작업의 효율성을 높이기 위한 다양한 지원을 제공한다.

본 논문에서 기술된 카드 시스템은 툴과 설계 환경이 구조적으로 결합되어 있는 집적형 카드 시스템에 속한다. 집적형 구조의 취약점인 시스템의 배타성을 개선하기 위하여 프레임워크의 툴 오브젝트 모델<sup>3)</sup>을 구조 집적에 적합하게 단순화하여 이용했다. 툴 오브젝트 모델은 다양한 툴의 수행 제어를 기능적으로 자동화하기 위해 복잡한 계층 구조(CTKO)와 지원 모듈(CAD tool modeller, Blackboard, interprocessor communication modellers, VLSI design methodology planner 등)을 필요로 한다. 또한 이 시스템은 사용자의 설계 작업에서 요구되는 지원 기능을 수행하는 모든 툴들을 포함하고 있어야 한다. 이 방법은 설계 자원이 제한되지 않은 방대한 시스템 구성을 위한 이질적인 설계 자원의 집적에는 효과적이다. 본 논문에서 설정한 제한 환경(Unix, X Window, PC 386, 공유 도메인의 카드 툴과 설



(a) 조합형

(b) 집적형

(c) 개방형

그림 9. 카드 시스템의 비교

Fig. 9. Comparative study of CAD systems.

계 환경 등)에 대한 보급 가능성을 기대하기는 어렵다. 본 논문에서는 톨 오브젝트에 대응되는 단층 구조 오브젝트를 집적형 카드 시스템의 톨 및 설계 환경 집적에 이용하였다. 카드 톨 오브젝트는 구조적으로 결합된 톨과 시스템간의 상호 작용을 처리하기 위한 루틴 정보를 갖고 있으며, 톨의 집적을 위해서는 오브젝트의 지원 루틴들을 지정된 기능으로 모듈화 구현하여 루틴 정보를 변수 테이블에 등록한다. 톨의 수행 제어를 위해서 오브젝트에 대한 ON/OFF 프로그래밍과 원형 순차 수행 방식을 이용하여 집적된 톨에 대한 효과적 제어를 도모했다. 설계 환경 집적을 위하여 단층 구조의 환경 오브젝트를 구성했는데, 각 설계 환경의 특성 정보를 환경 변수 테이블이 갖고 있다. 설계 환경의 특성 정보를 지정된 형태의 데이터 파일들로 구현하고 변수 테이블에 등록하면 설계 환경의 집적이 가능하다. 사용자는 오브젝트의 환경 이름과 단위 크기(Lambda)등을 프로그램하여 원하는 집적회로 설계를 위한 작업 환경을 시스템에 조성할 수 있으며, 설계 환경간 데이터 변환(계획 중)

등의 기능도 수행된다. 카드 프레임워크에서는 설계 환경에 대한 지원(조성, 변경등)을 수행하는 톨들이 요구되며, 수행 제어를 위한 톨 오브젝트의 프로그래밍이 필요하다.

3) 집적회로 설계 예

카드 시스템을 이용하여 LPC(Linear Predictive Coding) 음성 합성기 설계 작업이 진행 중에 있다. 시스템 데이터베이스의 계층 구조 모델을 이용하여 그림 10과 같은 LPC 회로의 라이브러리 계층 구조를 구성하였다. 각 라이브러리는 셀로 구성되며, 셀은 스키매틱 회로도인 퍼셋으로 구성되었다. 셀에 대한 퍼셋의 구성은 스키매틱과 레이아웃 견지에서 기술이 가능한데(그림 10), 현재 스키매틱 회로 퍼셋에 대한 준비가 거의 완료되었고 레이아웃 퍼셋을 설계할 예정이다. 셀 구성을 위한 스키매틱 퍼셋은 110개가 설계 완료되어 있으며 회로 구성의 기본 퍼셋(F/F, Latch 등)은 35개가 준비되어 있다. 스키매틱 회로는 스위치 레벨에서 설계되며, 회로 구현 방

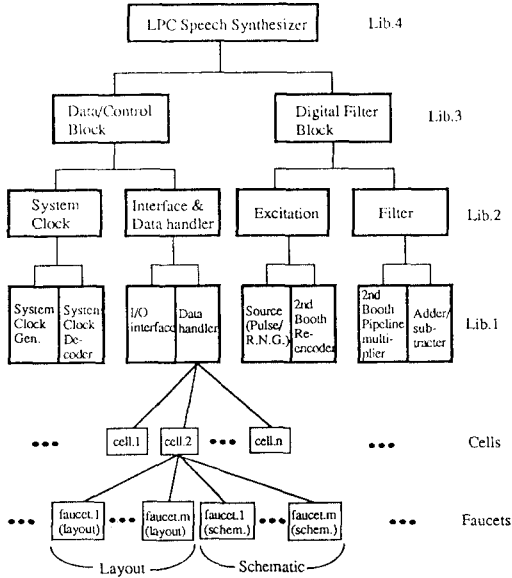


그림 10. LPC 음성 합성기 설계 데이터 계층 구조  
 Fig. 10. Hierarchical data structure of LPC Speech Synthesizer.

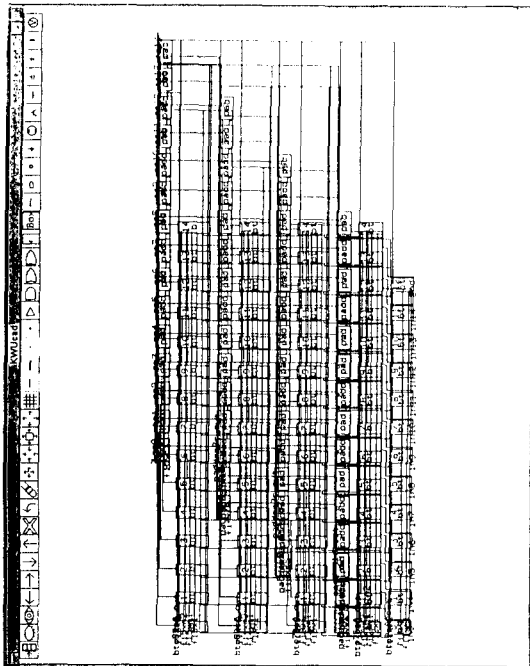


그림 11. LPC 음성 합성기의 15×10비트 2nd Booth 파이프라인 승산기  
 Fig. 11. 2nd Booth's 15×10bits pipeline multiplier in LPC Speech Synthesizer

법은 C2MOS(Clocked CMOS)를 이용하고 있다. 현재는 설계된 회로에 대한 검증 작업을 RSIM과 SPICE등 시뮬레이터를 통하여 진행하면서 검증된 셀을 구성하고 라이브러리 설계를 수행하고 있다. 라이브러리 계층 구조에서 부분적으로 회로 설계와 검증이 완료된 부분으로 시스템 클럭 발생기와 시스템 클럭 복호기(decoder) 및 2nd Booth's pipeline 승산기(multiplier)(그림 11)가 있다.

V. 결론 및 고찰

본 논문에서는 집적회로 설계용 집적형 카드 시스템의 설계 및 구현에 대하여 보았다. 카드 시스템은 실질적 표준 소프트웨어 개발 환경과 PC를 포함하는 중소형 컴퓨터 기종을 대상으로 시스템의 보급 가능성을 도모한다. 기존 집적형 시스템의 적응성 한계를 보완하기 위하여 카드 및 설계 환경 오브젝트와 변수 테이블을 제안 및 구현하여 카드 툴과 설계 환경에 대한 집적과 편집 가능성을 높였다. 또한 카드 툴 오브젝트 구조에 증분식 수행과 툴간의 통신을 고려하여, 설계 작업 중에 카드 시스템의 여러 툴이 동시에 지원을 제공할 수 있는 증분식 인터페이스를 구성했다. 설계를 진행하면서 카드 시스템의 구현 결함을 디버그하고 있으며 시스템 지원 기능 면에 대한 보안을 추진하고 있다. 현재 계획 및 진행되고 있는 주요 기능은 다음과 같다.

1) 설계 환경 변환기 및 편집기

레이아웃(CMOS, nMOS) 설계 데이터를 스키메틱(스위치, 게이트) 데이터로 변환시키는 기능을 추가하고 있다. 설계 환경의 편집을 위한 전용 편집기가 개발 중에 있다. 이 편집기는 설계 환경의 목차형 정보에 대한 변경 및 추가와 삭제 등의 기능을 수행할 수 있다.

2) VHDL(VHSIC Hardware Description Language) 입력 및 시뮬레이터

전자 시스템 기술을 위한 실질적 표준 언어인 VHDL에 대한 파서와 시스템의 구조 및 기능을 검증할 수 있는 시뮬레이션 방법이 개발되고 있다. VHDL 구조 시뮬레이션은 VHDL->RSIM 네트리스트 변환기와 RSIM 시뮬레이터로 수행되며, 기능 시뮬레이션은 VHDL->C 변환기와 C 컴파일러로 구현하려 한다.

3) 기존 캐드 툴에 대한 지원 확대  
현재 배선기와 PLA 발생기를 시스템에 탑재하는  
것을 계획하고 있다.

#### 參考 文獻

- [1] Steven M. Rubin, "An Integrated Aid for Top-Down Electric Design," *VLSI'83* pp.63-72, 1983.
- [2] H. Brown et al., "Palladio : An Exploratory Environment for Circuit Design," *IEEE Computer*, pp.41-56, DEC, 1983.
- [3] D.S. Harrison et al., "Data Management and Graphics Editing in the Berkely Design Environment," *Proc. of ICCAD*, pp.24-27, 1986.
- [4] J. Daniel, S.W. Director, "An Object Oriented Approach to CAD Tool Control within a Design Framework," *Proc. of 26th DAC*, pp.197-202, 1989
- [5] K.W. Fiduk, S.Kleinfeldt, et al., "Design Methodology Management - A CAD Framework Initiative Perspective," *Proc. of 27th DAC.*, pp.278-283, 1990.
- [6] Bill Johnson, "Bringing Design Management to the Open environment," *High Performance System*, pp.66-70, June, 1989.
- [7] C. Mead, L. Conway, *Introduction to VLSI system*, pp.47-51, 92-93, Addison-Wesley, 1980.
- [8] T.W. Griswold, "Portable Design Rules for Bulk CMOS.," *VLSI Design*, III-5, pp.62-67, Sept./Oct., 1982.
- [9] Amar Mukherjee, *Introduction to nMOS & CMOS VLSI System Design*, pp. 146, Prentice - Hall, 1986.
- [10] J.D. Ullman, *Computational Aspects of VLSI*, pp.268-269, Computer Science Press, 1984.
- [11] C.J. Terman, "RSIM - A LOGIC-LEVEL TIMING SIMULATOR," *Proc. of 1983 ICCD Conference*, pp.437-440, 1983.
- [12] R.E. Bryant, "A Switch-level Model and Simulator for MOS digital system," *IEEE Trans. on Computers*, Vol.C-33, No.2, pp.160-177, Feb., 1984.
- [13] R.E. Bryant, "COSMOS : A Compiled Simulator for MOS Circuits," *Proc. of 24th DAC.*, pp.9-16, 1987.
- [14] W.A. Christopher, *Spice3e1 manual*, UC-Berkeley, 1985.
- [15] Steven M. Rubin, *Computer Aids for VLSI Design*, Appendix B&D, Addison-Wesley, 1987.

## 著者紹介



孔 鎮 興(正會員)

1957年 9月20日生. 1980年서울 대학교 전자공학과 졸업(공학사). 1982年 한국과학 기술원 전기 및 전자 공학과 졸업(공학석사). 1989年미국 텍사스 주립대학(Austin) 대학원 전기 및 컴퓨터 공학과 졸업(공학 박사). 1982年~1986年 삼성전자 반도체 연구소 선임 연구원. 1989年~ 현재 광운대학교 컴퓨터 공학과 조교수. 주관심분야는 VLSI설계 자동화 툴 및 시스템 등임



金 載 協(正會員)

1969年 6月 16日生. 1992年2月 광운대학교 전자계산기 공학과 졸업(공학사). 1992年 3月 ~ 현재 광운대학교 대학원 전자계산기 공학과 석사과정 재학중. 주 관심분야는 VLSI CAD system, VHDL 등임



金 成 重(正會員)

1968年 10月 16日生. 1991年 2月 광운대학교 전자계산기 공학과 졸업(공학사). 1991年 3月~현재 광운대학교 대학원 전자계산기 공학과 석사과정 재학중(1993年 2月 석사 학위 취득 예정). 1992年 12月~ 현재 삼성휴렛팩커드 컴퓨터 개발실. 주관심분야는 VLSI CAD system 등임