

標準 電子回路 設計 데이터 形式 EDIF 紹介

張德浩 · 金 準 · 趙恩英 · 郭銘信 · 柳瑛昱
/ 自動設計應用研究室

〈要 約〉

CAD 워크스테이션과 설계도구가 다양해지고, 각 시스템이 고유한 설계 데이터 형식을 가짐에 따라 CAD 설계 데이터의 정보교환에 어려움이 생기게 되었다. 본고에서는 이러한 문제점을 해결하기 위해 최근 개발된 바 있는 EDIF란 표준전자회로 설계 데이터 양식에 관하여 소개한다. 내용으로는 EDIF의 개발 동기와 역사, 구문과 구조, 특성 및 미래에 관하여 언급하였다.

I. 서 론

워크스테이션간의 정보교환은 CAD/CAE 시스템 개발에 있어서 필수적이다. 집적회로의 설계 과정은 컴퓨터 설계 도구(Tool)와 silicon foundry 그리고 시스템 설계자 사이의 정보교환에 상당히 의존하고 있다. [2] 대부분의 IC는 워크스테이션에서 설계되는데,

일관성 없는 사용자 인터페이스, 임의 파일의 집합, 각 시스템이 고유한 형식을 가짐에 따라 정보 교환을 위해서는 데이터 번역의 필요성 등이 따른다.

과거 몇년간 silicon foundry와 워크스테이션 회사가 상당히 증가되었고 CAD/CAE 시스템도 급증하였는데, 이러한 시스템의 증가와 데이터의 교환 문제는 표준데이터 형식의 필요성을 절실하게 만들었다. CAD/CAE 시스템, silicon foundry, 시스템 설계자가 하나의 형식을 취하여 정보교환을 할 수 있게 되면 데이터 변환없이도 각각에 최적한 도구들을 선택하여 사용할 수 있게 될 것이고, CAD/CAE 산업과 silicon foundry 산업은 자유경쟁에 의해 더욱 발전될 수 있을 것이다. EDIF(Electronic Data Interchange Format)는 이러한 필요성에 부응하여 만들어진 것이다. 본고에서는 종래 데이터 형식의 문제점과 EDIF 발생 동기 및 역사, 구문과 구조, 특성 및 EDIF 미래에 관해 언급하였다.

II. EDIF 역사

그동안 많은 전자회로 설계 데이터 교환 형식과 하드웨어 기술언어(HDL)가 개발되었으나 각각은 아래와 같은 결점을 가지고 있었다.^(1,2,4)

- 좁은 범위 : 설계 데이터의 특정 양상만 언급함.
- 전유성 : 대부분의 형식은 특정 회사의 전유물이 됨.
- 구현의 난이 : 데이터 형식을 취급할 수 있는 parser / generator의 소프트웨어 개발이 어려움.
- 확장의 난이 : IC 설계 방법이 급속히 진보하고 있어, 데이터에 새로운 특성을 추가할 수 있어야 하나 이러한 요구를 만족시키기 위한 확장성이 없음.

EDIF는 기존의 주요한 4개의 데이터 형식의 특성을 하나의 강력한 형식으로 포함시키게 하기 위한 것으로, 각 형식의 개발자들이 모여서 1983년에 그 위원회가 구성되었다. 현재 이 위원회에 가입된 곳은 Daisy System, Mentor Graphics, Motorola, National Semiconductor, Tektronix, Texas Instrument, UC Berkeley의 7군데이며, EDIF가 기초한 주요 4가지 데이터 형식은 아래와 같다.^(1,2,4)

- CIDF (Common Interchange Description Format) : UC Berkeley의 연구결과를 중심으로한 데이터 형식으로 schematic 회로도, 심볼 레이아웃, 마스크 레벨 레이아웃, 연결도 등의 정보 전달을 목적으로 한 것이다. 구문은 LISP언어에 기초한 것으로 간단하고 확장성이 있다.
- GAIL (Gate Array Interface Language) : Daisy System 전용의 형식으로 설계의 특정 단계에서 게이트 어레이에 관련된 정보의 전달을 목적으로 한 것이다.
- TDF (Technology Definition File) : Motorola와 Mentor Graphics에 의해 공동 개발된 것으로 역시 게이트 어레이를 목

표로 하고 있다.

- TIDAL (Transportabl Integrated Design Automation Language) : Texas Instrument에서 개발된 데이터 형식으로 behavioral description에서 마스크 geometry까지 광범위한 설계 데이터를 기술할 수 있다. 테스트 데이터, 설계규칙, technology 데이터 등도 포함되는데 EDIF의 가장 큰 토대가 되었다.

EDIF 사양은 84년 5월에 비전 008로 처음으로 대중에게 공개되었고⁽¹⁾현재의 버전은 110이다. 주의해야 할 점은 EDIF는 설계 데이터의 전달을 위한 데이터 형식으로 프로그램 언어나 데이터 베이스 시스템이 아니라는 것이다. 이 데이터 형식은 설계 과정의 전 데이터를 기술해야만 하는 것은 아니고, 부분적으로만 기술해도 가능하다.

현재 EDIF가 지원할 수 있는 것은 semi-custom IC 설계와 게이트 어레이 설계이지만 PCB나 full custom digital IC, linear IC 등에도 사용될 수 있도록 확장하는 것이 고려되고 있다.

III. 구문과 구조

EDIF 구문의 기본 형식은 ASCII표현이며, EDIF 파일은 LISP 프로그램 언어에 기초한 계층적 구조로 되어 있다. 모든 데이터는 리스트, 즉 괄호내의 심볼릭 표현으로 나타난다. 각 리스트의 첫번째 요소는 키워드로서 특별한 의미를 갖는다. 이러한 구문은 매우 간결하여 쉽게 parsing될 수 있고, semantic weight는 키워드에 의해 생성되므로 확장이 가능하다는 장점이 있다. 새로운 키워드는 변화하는 수요에 의해 현존하는 parser의 변경 없이도 추가될 수 있고, 유사하게 구문이 변경되었을 때도 새로운 키워드는 무시되므로 기존 parser는 그대로 사용될 수 있다. 그러므로 구현하기가 간단하며 확장이 보장되는 것이다.

EDIF statement의 일반적 형태는 다음과 같다.

(keyword {form})

여기서 form은

symbolic constant/identifier/primitivedata/

EDIF statement

를 나타낸다. 키워드는 기억 용량을 줄이기 위해 단축하여 기술될 수 있다.

EDIF의 계층적 구조가 의미하는 것은 상위 레벨에서는 추상적이고 아래로 내려갈수록 점차 상세화되는 것을 의미한다. 최상의 레벨은 EDIF파일로서 한 설계에 필요한 모든 정보를 포함한다. 이 파일의 구성은 아래와 같다.

(EDIF ediffilename

status

{design}

{library}

{external}

{comment}

{user data})

Status block은 생성일시, 저자, 설계 기관과 수정 코드 등의 accounting 정보를 포함한다. 설계 section은 EDIF 기술의 시작점을 알려 주며, library block은 셀의 정의와 관련된 기술정보 등을 기술하며 external block은 파일에서 참조되었으나 현재 있지않은 라이브러리 리스트를 보여준다. User data 부분은 EDIF로는 기술될 수 없는 정보를 포함하는데 사용자가 부분적인 확장을 할 수 있게 한 부분이다.

EDIF의 가장 강력한 기능 중의 하나는 instantiation이다. 계층적 설계는 하나의 셀 정의내에 다른 셀을 instance함으로써 생성되는데 라이브러리나 셀이 그 자체가 nested되게 할 수는 없다. EDIF의 셀의 계층은 아래와 같다.

(LIBRARY LIBRARYNAME)

(STATUS)

(WRITTEN)

(AUTHOR "D. H. Chang")

(TECHNOLOGY CELLI. . .)

(CELL. . .)

(CELL. . .)

(CELL. . .)

각 셀들은 여러 형태의 데이터를 포함할 수 있는데 데이터의 유사한 형태를 관련짓기 위해서 view라는 개념을 사용한다. 하나의 셀은 제로 혹은 그 이상의 view를 포함할 수 있는데 각 view는 같은 셀의 다른 특성을 나타낸다. 예를들면 레이아웃 규칙 검증 프로그램은 단지 마스크 레벨의 데이터에만 관심이 있을 것이며, 회로 시뮬레이션 프로그램은 넷 리스트와 관련된 parasitic 정보만을 필요로 할 것이다. 현재의 버전에서는 아래의 view가 제공된다.

- mask layout
- net list
- schematic
- symbolic
- behavior
- document
- stranger

Mask layout view는 geometry와 geometry 기술 (Description)이 어떻게 번역되고 화면에 나타날 것인가에 관한 정보를 포함하며 연결도에 관한 정보는 포함하고 있지 않다.

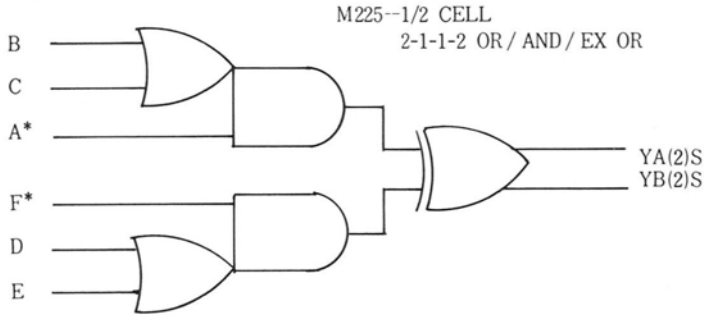
넷 리스트, schematic, 심볼릭 view는 연결도와 그래픽 정보를 포함한다. Schematic view는 논리 다이어그램을, 심볼릭 view는 심볼릭 layout을 나타내고, 넷 리스트 view는 연결도 정보를 나타낸다.

Behavior view는 동작기능과 시뮬레이터 입력 데이터와 같은 테스트 데이터를 위해 사용되며 document view는 설계의 문서화를 위해 다이어그램과 글자를 포함하고 있다. 마지막으로 stranger view는 다른 모든 view에 적합하지 않는 셀에 관한 데이터를 포함하고 있다.

하나의 view는 또한 인터페이스 부분과 내

용 (Contents) 부분의 두가지 부분으로 나누어 질수 있다. 인터페이스 부분은 다른 셀과의 인터페이스 정보를 가지고 있으며 내용 부분은 다른 셀의 instantiation 정보를 포함한 실제 셀의 구현에 필요한 정보를 담고 있다.

〈그림 1〉은 Motorola의 Macrocell M225를 나타내고 〈그림 2〉는 그에 대응하는 schematic view의 EDIF 표현을 나타낸다. EDIF의 mask layout view와 CIF⁽⁵⁾의 대비를 위해 〈그림 3〉에 cifExample이란 셀을 root로 한 두 가지의 description을 예시하였다.



〈그림 1〉 Motorola M225 Macrocell

```

(EDIF SCHEMATICEXAMPLE
 (STATUS (EDIFVERSION 1 0 0)(EDIFLEVEL 0)
 (WRITTEN (TIMESTAMP 1985 1 21 14 33 10))(COMMENT "End Written")
 (COMMENT "End Status")(COMMENT "Sample library for EDIFSCHM macro cells")
 (LIBRARY LIB1
 (TECHNOLOGY SCHEMATIC__SYMBOL
 (NUMBERDEFINITION ENGLISH (SCALE DISTANCE 1 (E 10 -3))))
 (COMMENT "EDIF unit is 10 mils")
 (COMMENT
 "This library has been created by inserting the above
 header with the output from EDIFSCHM"
 )
 (CELL M225
 (STATUS (EDIFVERSION 1 0 0)(EDIFLEVEL 0)
 (WRITTEN (TIMESTAMP 1985 05 07 17 05 07)
 (ACCOUNTING AUTHOR "HART SWITZER, RRTW60")
 (ACCOUNTING PROGRAM "EDIFSCHM VERS M0702.00.001.000
 ***03/14/85***))
 (VIEW SCHEMATIC MACRO__SCHEMATIC
 (INTERFACE (DEFINE INPUT PORT (MULTIPLE B C A F D E)
 (DEFINE OUTPUT PORT (MULTIPLE YA YB))
 (PORTIMPLEMENTATION B (FIGUREGROUP POINT (DOT (POINT 82 672))))
 (PORTIMPLEMENTATION C (FIGUREGROUP POINT (DOT (POINT 85 625))))
 (PORTIMPLEMENTATION A (FIGUREGROUP POINT (DOT (POINT 85 568))))
 (PORTIMPLEMENTATION F (FIGUREGROUP POINT (DOT (POINT 85 429))))
 (PORTIMPLEMENTATION D (FIGUREGROUP POINT (DOT (POINT 89 377))))
 (PORTIMPLEMENTATION E (FIGUREGROUP POINT (DOT (POINT 85 336))))
 (PORTIMPLEMENTATION YA (FIGUREGROUP POINT (DOT (POINT 745 540))))
 (PORTIMPLEMENTATION YB (FIGUREGROUP POINT (DOT (POINT 745 453))))
 (BODY
 (USERDATA BORDER MACRO__BORDER
 (FIGUREGROUP BORDER (BORDERPATTERN 6 "001111")
 (RECTANGLE (POINT 109 227)(POINT 718 774))))))
 (CONTENTS
 (FIGUREGROUP SCHEMATIC__SYMBOL (FILLPATTERN 1 1 "0")
 (PATH (POINT 438 407)(POINT 465 407))
 (PATH (POINT 465 407)(POINT 465 484))
 (PATH (POINT 465 484)(POINT 492 484))
 (PATH (POINT 492 530)(POINT 464 530))
 (PATH (POINT 464 530)(POINT 464 601))
 (PATH (POINT 464 601)(POINT 437 601))
 (PATH (POINT 78 676)(POINT 86 668))
 (PATH (POINT 86 676)(POINT 78 668))
 (PATH (POINT 82 672)(POINT 187 672))
 (PATH (POINT 81 629)(POINT 89 621))
 (PATH (POINT 89 629)(POINT 81 621))
 (PATH (POINT 85 625)(POINT 190 625))
 (PATH (POINT 81 572)(POINT 89 564))
 (PATH (POINT 89 572)(POINT 81 564))

```

```

(PATH (POINT 85 568)(POINT 319 568))
(PATH (POINT 81 433)(POINT 89 425))
(PATH (POINT 89 433)(POINT 81 425))
(PATH (POINT 85 429)(POINT 319 429))
(PATH (POINT 85 381)(POINT 93 373))
(PATH (POINT 93 381)(POINT 85 373))
(PATH (POINT 89 377)(POINT 190 377))
(PATH (POINT 81 340)(POINT 89 332))
(PATH (POINT 89 340)(POINT 81 332))
(PATH (POINT 85 336)(POINT 190 336))
(PATH (POINT 741 544)(POINT 749 536))
(PATH (POINT 749 544)(POINT 741 536))
(PATH (POINT 745 540)(POINT 612 540))
(PATH (POINT 741 487)(POINT 749 479))
(PATH (POINT 749 487)(POINT 741 479))
(PATH (POINT 745 483)(POINT 620 483))
(PATH (POINT 318 356)(POINT 388 356))
(PATH (POINT 318 356)(POINT 318 456))
(PATH (POINT 318 456)(POINT 388 456))
(SHAPE (ARC (POINT 388 356)(POINT 438 406)(POINT 388 456)))
(PATH (POINT 318 550)(POINT 388 550))
(PATH (POINT 318 550)(POINT 318 650))
(PATH (POINT 318 650)(POINT 388 650))
(SHAPE (ARC (POINT 388 550)(POINT 438 600)(POINT 388 650)))
(PATH (POINT 178 305)(POINT 231 305))
(PATH (POINT 178 405)(POINT 231 405))
(SHAPE (ARC (POINT 178 305)(POINT 191 355)(POINT 178 405)))
(SHAPE (ARC (POINT 231 305)(POINT 281 318)(POINT 318 355)))
(SHAPE (ARC (POINT 231 405)(POINT 281 392)(POINT 318 355)))
(PATH (POINT 178 597)(POINT 231 597))
(PATH (POINT 178 697)(POINT 231 697))
(SHAPE (ARC (POINT 178 597)(POINT 191 647)(POINT 178 697)))
(SHAPE (ARC (POINT 231 597)(POINT 281 610)(POINT 318 647)))
(SHAPE (ARC (POINT 231 697)(POINT 281 684)(POINT 318 647)))
(PATH (POINT 501 458)(POINT 554 458))
(PATH (POINT 501 558)(POINT 554 558))
(SHAPE (ARC (POINT 481 458)(POINT 494 508)(POINT 481 558)))
(SHAPE (ARC (POINT 501 458)(POINT 514 508)(POINT 501 558)))
(SHAPE (ARC (POINT 554 458)(POINT 604 471)(POINT 641 508)))
(SHAPE (ARC (POINT 554 558)(POINT 604 545)(POINT 641 508)))
(ANNOTATE "B" (POINT 120 682)(POINT 129 702)(LOWERLEFT)
(ANNOTATE "C" (POINT 120 630)(POINT 129 650)(LOWERLEFT)
(ANNOTATE "A(*)" (POINT 120 578)(POINT 165 598)(LOWERLEFT)
(ANNOTATE "F(*)" (POINT 120 434)(POINT 165 454)(LOWERLEFT)
(ANNOTATE "D" (POINT 120 382)(POINT 129 402)(LOWERLEFT)
(ANNOTATE "E" (POINT 120 346)(POINT 129 366)(LOWERLEFT)
(ANNOTATE "$ (2) YA" (POINT 706 550)(POINT 634 570)(LOWERRIGHT)
(ANNOTATE "$ (2) YB" (POINT 706 488)(POINT 634 508)(LOWERRIGHT)
(ANNOTATE "M225--1/2CELL" (POINT 315 723)(POINT 459 743)
(LOWERLEFT))))))

```

〈그림 2〉 Motorola M225 Macrocell의 EDIF Description

```
(CIF file of symbol hierarchy rooted at cifExample);
DS 2 1 1;
9 MPContact;
L NP;
B 800 800 0 0;
L NM;
B 1200 1200 0 0;
L NC;
B 400 400 0 0;
DF;
DS 1 1 1;
9 cifExample;
C 2 T -1000 1000;
C 2 T 2000 -2000;
L NM;
W 600 2000 -2000 2000 1000 -1000 1000;
DF;
C 1;
E
```

```
design cifExample (qualify cellLib cifExample)
comment "EDIF description of symbol hierarchy rooted at cifExample*"
library cellLib
(technology nmosMC (comment "see below"))
:cell MPContact
(view maskLayout Physical
(contents
(figureGroup NP (rectangle (point -400 -400) (point 400 400)))
(figureGroup NM (rectangle (point -600 -600) (point 600 600)))
(figureGroup NC (rectangle (point -200 -200) (point 200 200)))
)
)
)
:cell cifExample
(view maskLayout Physical
(contents
(instance MPContact Physical MPC1 (transform (translate -1000 1000)))
(instance MPContact Physical MPC1 (transform (translate 2000 -2000)))
(figureGroup NM (path (point 2000 -2000) (point 2000 1000)
(point -1000 1000)))
)
)
)
```

〈그림 3〉 CIF와 EDIF의 Cell Description

IV. EDIF의 미래

EDIF는 사용자들의 응답을 반영시켜 가면서 계속적으로 진보되어 가고 있다. 개발중에 있는 분야로서는 테스트 프로그램 기술, functional, behavior, 논리 모델 동작의 정의, 레이아웃 설계규칙, 전기적 설계규칙, 패키지 기술, PCB 레이아웃 등이 있다.

EDIF와 관련되어 필요한 소프트웨어는 schematic을 EDIF로 바꾸어주는 번역 프로그램과 셀의 정확성을 검증하기 위한 plot 프로그램 등이 있다. 이러한 소프트웨어는 사용자가 EDIF를 모르더라도 충분한 EDIF 기술이 가능하도록 해 줄 것이다.

EDIF는 점차로 많은 CAE/CAD 시스템 공급자들에 의해 채택되어 지고 있으며,^[6] CAD 데이터 베이스 설계에서도 이용되고 있다.^[7]

Motorola에서는 게이트 어레이 라이브러

리를 customer에게 제공하는데 EDIF를 사용할 계획을 가지고 있으며 8명의 EDIF 전담 인원을 두고 있다.

현재 문제점으로 지적되고 있는 것은 시뮬레이션 모델에 관한 일치가 없고, Lisp와 유사한 syntax와 ASCII 표현은 계산상의 overhead와 저장 용량의 증대를 가져온다는 점 등이 있다.^[3]

EDIF 외에도 설계 데이터를 표준화하려는 노력은 몇가지가 있다. IGES (Initial Graphics Exchange Specification)는 NBS (National Bureau of Standard)의 지원을 받고 있는데 공통 CAD 인터페이스 언어를 개발하기 위한 것이고, 그외에도 board-level CAD 출력 형식을 개발하려는 노력과 하드웨어 기술 언어를 개발하려는 노력이 있어, EDIF가 PCB 레이아웃까지 제공할 수 있게 된다면 EDIF와 이러한 다른 양식간에 어느 정도의 중복성이 생길 것이다.

EDIF 위원회는 사용자들의 의견을 수렴하고, EDIF의 보급과 정보 전달을 위해 EDIF user group을 운영하고 있는데, 그 주소는 아래와 같다.

The EDIF User Group
Design Automation Department,
Texas Instrument, PO Box 225474, MS3668,
Dallas, TX75265.

V. 결 론

CAD와 반도체, PCB 기술의 발전에 따라 CAD 워크스테이션과 설계 도구의 종류가 급증하였으나, 각 시스템들은 각기 다른 데이터 형식을 가짐에 따라 일일이 번역 프로그램이 필요하는 등 설계 정보의 호환성에 큰 문제를 가져왔다. 본고에서는 이러한 문제점을 해결하기 위한 전자회로 설계의 표준 데이터 형식인 EDIF의 출현 배경과 역사, 구문과 구조, 특성 및 미래에 관하여 기술하였다.

EDIF는 아직도 발전중에 있는 데이터 형

^[1] 식이므로 이의 기술적 발전 추세를 충분히 검토하고, 세계적인 EDIF 이용 현황들을 지켜본 후 이의 실제적인 활용이 가능할 것이다. 이를 활용하기 위해서는 각 설계 도구의 출력 형식을 EDIF에 맞게 통일시키거나 기존 데이터 형식을 변환하는 번역 프로그램이 필요하게 된다. EDIF는 데이터 형식의 표준화 뿐 아니라 CAD 데이터베이스의 스키마 설계에도 이용될 수 있을 것으로 보인다.

〈参 考 文 献〉

1. EDIF Steering Committee, EDIF Electronic Design Interchange Format Version 110, 1985.
2. Esther R. Marx, "EDIF : The Standard for Workstation Intercommunication"
3. Richard Goering, "Design Tool Vendors Move toward Integrated Systems" Computer Design, June, 1985.
4. John D. Crawford, "EDIF : A Mechanism for the Exchange of Design Automation" IEEE, 1984.
5. C. A. Mead, L. A. Conway, Introduction to VLSI Systems, Addison-Wesley, 1980.
6. Hung-fai Stephan Law, "Workstations: A Complete Solution to the VLSI Designer ?," Proc. of 22nd DAG, ACM /IEEE, 1985.
7. D. S. Batory W. Kim, "Modeling Concepts for VLSI CAD Objects, ACM Trans. on Database System, Vol. 10, No. 3, Sept., 1985.