

EDAS_P 시스템에서의 Netlist 抽出方法 (SCHEX_P)

朴仁學 · 李哲東 · 柳瑛昱

〈要 約〉

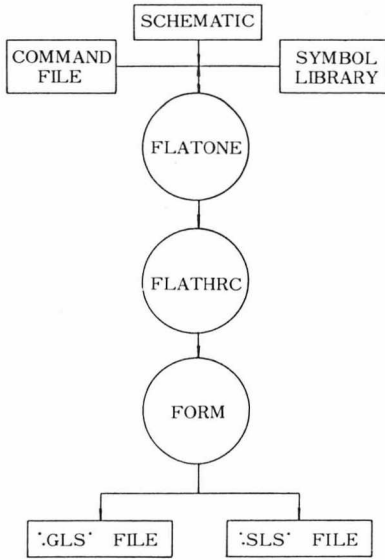
전자회로를 시뮬레이션하려면 게이트나 트랜지스터의 연결상태(netlist)를 입력시켜야 한다. EDAS_P시스템의 SCHEX_P라는 tool은 그림으로 표현된 전자회로도로부터 연결상태를 추출하고, 계층 설계된 구조를 풀어 게이트나 트랜지스터만으로 표현된 netlist를 만든 후 시뮬레이터가 받아들일 수 있는 형식으로 문장을 재조립한다. 본고에서는 이 과정을 요약하여 설명하고자 한다.

I. 서 론

Netlist란 전자회로의 topology, 즉 회로의 연결상태를 나타내는 수단으로서 시뮬레이션의 입력 파일에는 예외없이 netlist라는 것을 만들어

주어야 한다. 종래에는 pen으로 회로를 종이에 그리고 wire에 번호를 매겨 netlist를 구해 컴퓨터에 일일이 입력시키는 방식을 취해왔으나 작업이 매우 단조롭고 지루한 과정이라 실수가 발생할 확률도 많았다. 본 EDAS_P 시스템의 SCHEX_P(SCHematic EXtractor - Personal)는 이 과정을 자동화시켜 netlist를 자동으로 추출하는 것이 가능하도록 하였다.

이 tool에서 extraction 과정은 3단계로 나누어져 3개의 프로그램이 자동으로 연결되어 수행되도록 되어있다. 〈그림 1〉의 flowchart에 나타나 있는 FLATONE이라는 프로그램은 schematic의 그림 정보로부터 netlist를 추출하는 것이고, FLATHRC라는 프로그램은 netlist간의 hierarchy를 풀어 flat 한 netlist를 얻는 프로그램이다. FORM라는 프로그램은 추출된 netlist를 시뮬레이터가 받아들일 수 있는 형태로 syntax를 재조립하는 역할을 한다.



〈그림 1〉 회로 추출과정의 흐름도

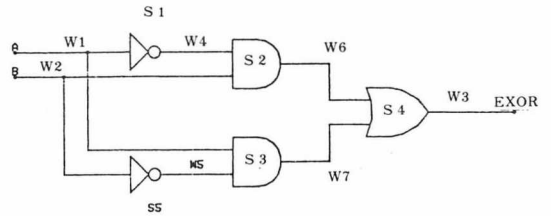
II. 본 론

1. Netlist

Netlist란 회로의 topology를 나타내는 수단으로서 게이트나 트랜지스터가 어떻게 연결되었는가를 고유이름을 이용해 표현한다. 예를 들어 2 input exclusive OR 회로를 게이트를 이용해 만들면 〈그림 2〉과 같다. 각각의 게이트에는 고유한 이름으로서 “S”로 시작하여 번호를 붙인 S1 ~ S5까지의 이름으로 나타내고, 연결된 wire는 같은 이름을 갖고 떨어진 wire는 다른 이름을 갖도록 “W”로 시작하며 일련의 번호를 붙이면 W1 ~ W7까지 나타낼 수 있다. 이 고유이름을 이용해 netlist라는 것으로 표현하는데 게이트의 고유이름을 맨 처음에 적은 후 게이트의 output에 연결된 wire의 이름을 쓰고, 다음 게이트의 기능을 나타내는 단어를 적은 후 입력단자에 연결된 wire의 이름을 차례로 나열하는 형식을 취하면 아래와 같은 netlist를 얻는다.

S1 W4 INV W2
S2 W6 AND W4 W2

S3 W7 AND W1 W5
S4 W3 OR W6 W7
S5 W5 INV W2

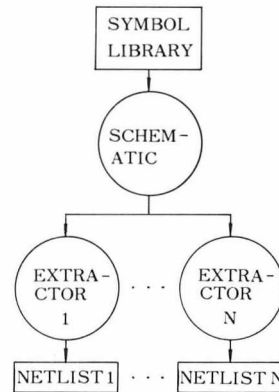


SCHEMATIC FILENAME : EXOR. SCH

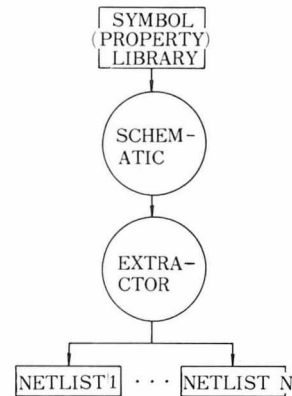
〈그림 2〉 Gate 로 표현한 2 Input EXOR 회로도

2. Netlist 추출 방법 비교

시뮬레이터마다 입력되는 netlist의 표현방식



(a) 종래의 CAE 시스템에서의 Extraction 방식



(b) EDAS_P 시스템에서의 Extraction 방식

〈그림 3〉 Extractor 의 구조 비교

이 다르고 또 필요로 하는 정보도 차이가 난다. 종래의 CAE 시스템은 추출하는 netlist에 따라 별도의 extractor가 만들어져 있어 추출하기 원하는 netlist에 따라 별도의 extraction tool을 사용해야 한다(〈그림 3(a)〉 참고). 그러나 EDAS_P 시스템에서는 symbol 정의시 symbol에 property를 정의하여 netlist의 형식을 정의함으로써 extractor는 하나를 사용하지만 추출할 수 있는 netlist의 형식은 여러가지가 될 수 있도록 구현시켰다(〈그림 3(b)〉 참고).

3. Property

EDAS_P 시스템은 symbol에 property라는 것을 정의함으로써 그 symbol의 특징 및 netlist를 구성하는 형식을 정의한다.

가. 사용할 시뮬레이터의 명시

Netlist는 시뮬레이터의 입력 파일로 사용할 것이므로 시뮬레이터의 이름을 명시하여 뒤따라오는 문장을 어느 시뮬레이터의 netlist를 추출할 때 사용할 것인가를 알려준다. 형식은 대문자로 나타낸 시뮬레이터의 이름에 곧바로 > 를 붙인다.

Simulator name >

예) GLSIM_P >, SLSIM_P >

나. Symbol 종류의 명시

- User defined symbol(default)
계층설계시 사용자가 정의한 symbol로서 property를 정의하지 않는다.

- Primitive symbol
해당 시뮬레이터에서 최하단의 level인 primitive임을 알려준다.

PRIMITIVE라는 고유어를 사용하며 표현 형식은 아래와 같다.

Simulator_name >PRIMITIVE

- Connector Symbol

이 symbol에 연결된 wire는 hierarchy에 상

관없이 연결된 wire에 일정한 이름을 부여하는 역할을 하는 symbol로서 power나 ground선의 이름을 정의하는 데 유용하다.

Simulator_name >CONTSYM : connector_name wire_name

예) SLSIM_P >PRIMITIVE

SLSIM_P >CONTSYM : VDD Vdd

다. 문장 구성 형식의 정의

시뮬레이터의 입력 파일의 형식을 정의하는 문장으로써 FORMAT이라는 고유어를 사용하고 뒤따라나오는 형식문과는 colon(:)으로 구분한다.

simulator_name >FORMAT : 형식문

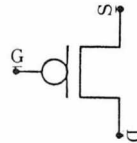
- 고유어 : '와 '사이에 넣는다.

- Symbol의 identity name : @S

- Symbol connector에 연결된 wire의 identity name : symbol의 connector name

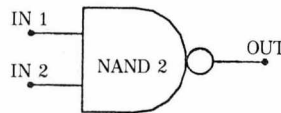
- Space : 문장에 있는 그대로 나타남

2-input NAND gate에 정의된 property와 P type transistor에 정의된 property가 〈그림 4〉에 나타나 있다.



SLSIM_P >PRIMITIVE

SLSIM_P >FORMAT: 'P' G S D



GLSIM_P >PRIMITIVE

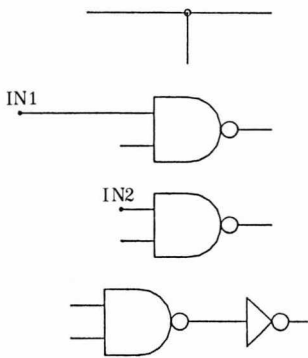
GLSIM_P >FORMAT: OUT 'NAND 4' IN1 IN2

〈그림 4〉 Property의 예

라. 연결이 검사되는 경우

Schematic에서 연결이 검사되는 도형은 schematic의 connector와 wire, 그리고 symbol의 connector이다. 이들은 상호간의 연결상태가 검사되어 netlist를 추출하게 된다.

- Wire간의 연결상태
- Wire가 schematic connector나 symbol connector에 연결된 경우
- Symbol connector 위에 schematic connector가 배치된 경우
- Symbol connector가 서로 연결된 경우



〈그림 5〉 연결이 이루어지는 경우

마. 추출된 netlist의 일반형식

```

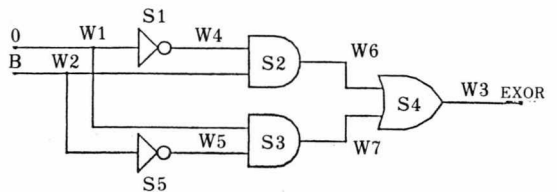
SCHEMATIC FILENAME , NUMBER OF CONNECTOR
CONNECTOR NAME , WIRE IDENTITY
.
.
.
CONNECTOR NAME , WIRE IDENTITY
SYMBOL FILENAME , SYMBOL IDENTITY
NUMBER OF CONNECTOR , 0
CONNECTOR NAME , WIRE IDENTITY
.
.
.
CONNECTOR NAME , WIRE IDENTITY
[
.
.
.
]
END
    
```

〈그림 6〉 Netlist의 형식

시뮬레이터에 따라 netlist의 형식이 맞추어지기 전, 일반적인 형식으로 추출되는 netlist는 〈그림 6〉과 같은 형식을 갖는다. 처음에 schematic이 갖고 있는 connector 이름과 연결된 wire name이 나열되고 다음은 symbol마다의 이름과 symbol connector에 연결된 wire name이 나열된다.

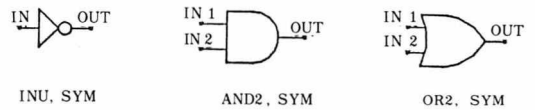
바. Schematic에서 netlist의 추출

〈그림 7(a)〉에 나타난 2-input exclusive OR 회로는 inverter와 2-input AND, 2-input OR gate로 이루어지므로 1개의 schematic file (EXOR. SCH)과 3개의 symbol file(INV. SYM,



SCHEMATIC FILENAME : EXOR. SCH

(a) 2 input EXOR 회로의 schematic



INV. SYM AND2. SYM OR2. SYM

(b) EXOR 회로에 사용된 symbol들

```

EXOR , 3
A , W1
B , W2
EXOR , W3
[
INV , S1
2 , 0
IN , W1
OUT , W4
]
[
INV , S5
2 , 0
IN , W2
OUT , W5
]
[
AND2 , S2
3 , 0
IN1 , W4
IN2 , W2
OUT , W6
]
[
AND2 , S3
3 , 0
IN1 , W1
IN2 , W5
OUT , W7
]
[
OR2 , S4
3 , 0
IN1 , W6
IN2 , W7
OUT , W3
]
[
AND2 , S4
3 , 0
END , END
]
    
```

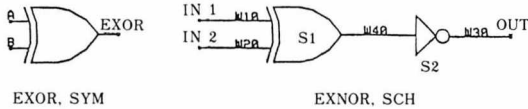
(c) EXOR 회로의 netlist

〈그림 7〉 2 Input EXOR 회로의 Netlist

AND2. SYM., OR2. SYM)로부터 netlist를 추출하면 <그림 7(c)>에 나타난 netlist와 같은 결과를 얻는다.

사. Flattening에 의한 netlist 추출

<그림 2>의 2-input exclusive OR 회로에 해당하는 symbol을 <그림 8(a)>와 같이 정의하고 2 input exclusive NOR 회로를 그림 8(b)와



(a) 2input EXOR 회로의 symbol (b) 2input EXNOR 회로의 symbol

```

EXNOR , 3
IN1 , W10
IN2 , W20
OUT , W30

[ EXOR , S1
  3 , 0
  A , W10
  B , W20
  EXOR , W40

[ INV , S2
  2 , 0
  IN , W40
  OUT , W30

END , END

MAX. WIRE NUMBER=W40
MAX. SYMBOL NUMBER=S2
    
```

(c) EXNOR 회로의 netlist

```

EXOR , 3
A , W1 W10
B , W2 W20
EXOR , W3 W40

[ INV , S1 S3
  2 , 0
  IN , W1 W10
  OUT , W4 W41

[ INV , S5 S4
  2 , 0
  IN , W2 W20
  OUT , W5 W42

[ AND2 , S2 S5
  3 , 0
  IN1 , W4 W41
  IN2 , W2 W20
  OUT , W6 W43

[ AND2 , S3 S6
  3 , 0
  IN1 , W1 W10
  IN2 , W5 W42
  OUT , W7 W44

[ OR2 , S4 S7
  3 , 0
  IN1 , W6 W43
  IN2 , W7 W44
  OUT , W3 W40

END , END
    
```

(d) Flattening의 과정

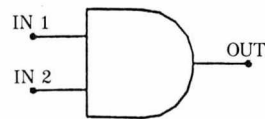
<그림 8> 2-input EXOR 회로의 Netlist Flattening

같이 정의했을때, 이 회로도로부터 netlist를 추출하면 <그림 8(c)>와 같이 얻을 수 있지만 EXOR gate는 primitive가 아니므로 <그림 7(c)>의 결과와 연관지어 hierarchy를 푸는 작업이 이루어져야 한다. <그림 8(d)>의 EXOR 회로도의 netlist에서 EXOR schematic의 connector인 A, B, EXOR에 연결된 wire의 이름은 W1, W2, W3이고, EXNOR schematic에서 EXOR symbol의 connector인 A, B, EXOR에 연결된 wire의 이름은 W10, W20, W40이다. 같은 이름의 단자에 연결된 wire의 이름을 같게 하기 위해 W1을 W10으로, W2을 W20으로, W3을 W40으로 바꾸어 준 후 나머지 wire들의 번호를 EXNOR schematic의 wire 번호와 겹치지 않도록 renumbering을 하고, EXOR symbol의 netlist 대신 schematic의 netlist로 바꾸면 flattening이 이루어진다.

아. 문장의 재구성

앞서 설명한 과정으로 추출되고 flattening된 netlist는 symbol이 갖는 property에 따라 문장이 재구성된다.

- 2-input AND gate의 property가 아래와 같고



```

GLSIM_P>PRIMITIVE
GLSIM_P>FORMAT: OUT 'AND 1
1' IN1 IN2
    
```

- Flatten된 netlist의 일부분이 아래와 같다고 하면

```

AND2 , S5
3 , 0
IN1 , W41
IN2 , W20
OUT , W43
    
```

- Property의 FORMAT에 정의된 문장구성법에 따라 OUT는 OUT라는 symbol

connector에 연결된 wire이름인 W43이 대치되고, 다음 space후 고유어 AND 1 1 이 따라붙고, 계속해 IN1과 IN2라는 문자는 해당 symbol connector에 연결된 wire이름으로 대치된다. 그 결과가 아래에 나타나있다.

W43 AND 1 1 W41 W20

Ⅲ. 결 론

SCHEX_P는 netlist를 추출하는 과정을 세 단계로 나누어 하나는 top schematic으로부터 primitive level까지 사용자가 만든 모든 schematic의 netlist를 추출하는 과정이고 다른 단계는 netlist간의 hierarchy를 푸는 과정이다. Netlist를 시뮬레이터의 입력 파일로 조립하는 과정이 마지막 단계로써 이 과정을 마치면 완전한 시뮬레이터 입력 파일을 얻는다.

본 시스템의 netlist 추출방식은 extractor의 종류에 의한 추출이 아닌 symbol property의 정의에 의한 추출이기 때문에 다른 형태의 netlist를 얻는 것이 매우 용이하도록 설계되어 있다.

〈參考文獻〉

1. T. C. Raymond, "LSI/VLSI Design

Automation, "VLSI Support Technologies: CAD, Testing, and Packaging, IEEE Computer Society, pp.1982, 134~146.

2. J. C. Foster, "A Unified CAD System for Electronic Design", ACM IEEE 21st. Design Automation Conf., pp. 365~369, 1984.

3. A. R. Newton, A. L. Sangiovanni - Vincentelli, "Computer - Aided Design for VLSI Circuits," IEEE Computer, vol. 19, no. 4, pp. 38~59, April, 1986.

4. VLSI Systems Design Staff, "Survey of CAE systems," VLSI Systems Design, vol. VII, no.6, pp. 85~105, June 1986.

5. Technology to Watch, "Turning a PC into a Silicon Compiler," Electronics, pp. 37~40, June 1986.

6. 유 영욱외 44명, "설계 자동화 시스템 개발에 관한 연구", 1985년도 과학 기술처 특정 연구사업, 최종 보고서, 한국전자통신연구소, 1986. 7.

7. EDAS-P 운용 기술서, 참고 문헌[6]의 별책 부록, 1986. 7.

8. 박 인학, 이 철동, 유 영욱, "전자 회로. 설계를 위한 개인용 CAE 시스템", 전자공학회 1986년도 CAD·반도체·재료 및 부품 연구회 합동학술 발표회 논문집, pp. 11~14, 1986. 5.

朴仁學* 18 페이지 참조
李哲東* 24 페이지 참조
柳瑛昱* 11 페이지 참조