

## 마크로모델 개발을 위한 통합 시스템

### (An Integrated System for Macromodel Development)

朴 璉 圭\*, 鄭 義 英\*, 金 敬 昊\*

(Jin Kyu Park, Eui Young Chung and Kyung Ho Kim)

#### 要 約

본 논문에서는 마크로 모델(macromodel)을 쉽게 개발하기 위한 새로운 시스템인 BEST(Behavior Evaluation Supporting Tool)에 대해 기술한다. BEST에서는 방정식과 스펙이 주어진 임의의 마크로 모델에 대해서 회로의 component 값을 계산하며, 특정 온도 및 바이어스 조건하에서 트랜지스터 레벨 회로에 대한 마크로 모델 회로의 결과를 비교 분석하여 모델 개발에 필요한 정보를 제공하고, 파라미터 값을 최적화시켜 준다. 또한, CMOS 논리 회로에 사용되는 MOSFET을 PWL-RC(Piecewise Linear-RC) 네트워크로 구성되는 스위치 모델로 특성화할 수 있기 때문에 마크로 / 스위치 / 트랜지스터 레벨이 혼재된 멀티레벨 시뮬레이션이 가능하여 시뮬레이션 속도와 정확도간의 trade-off를 할 수 있다. BEST는 다양한 그래픽 사용자 인터페이스를 통해 위와 같은 마크로 모델 개발 환경을 제공함으로써 설계자가 하기 어려웠던 마크로 모델 개발을 직접 수행하여 활용할 수 있도록 하였다. 실험 회로에 대해 BEST를 적용한 결과 마크로 레벨의 경우 18.6배의 시뮬레이션 속도 향상을 가져왔고 10% 내외의 정확성을 유지하였으며, 최적화 기능의 유용성을 검증하였다.

#### Abstract

In this paper, we describe a new system, called BEST, that is used to develop a macromodel or behavioral model easily. It automatically calculates the component values of macromodel represented by equations to satisfy the given specification. Also, it gives the way to analyze both the behavioral model and transistor level circuit, and then compare the analysis results of them to check the correspondence under specific temperature and bias condition, and BEST optimizes the component values of macromodel. Other feature is to characterize MOSFET as switch model which consists of PWL-RC network. Finally, it is possible to generate multi-level netlist which consists of macro / switch / transistor level circuits, and user can determine the trade-off between simulation speed and accuracy. With the graphic user interface form of macromodel development system described above, BEST enable designers to make macromodel by themselves and to use it. We applied BEST to develop the macromodel for the test circuit and got the 18.6 times simulation speed up with preserving the accuracy within 10% compared to the conventional transistor level circuit simulation. Also, applicability of optimization capability was verified.

## 1. 서론

반도체 기술의 발달로 VLSI 회로의 집적도가 높아짐에 따라 회로의 규모가 급격히 증가하게 되어 설계검증의 필수적인 과정인 회로 시뮬레이션 수행시간이 기하급수적으로 증가하게 되었다. 이에 대한 해결책으로, 시뮬레이터 자체 알고리즘 개선을 위한 노력이 진행되고 있으며, 또다른 해결책으로서 매크로 / 행위 모델링 기법(macro / behavioral modeling method)이 제시되고 있는 상황이다.<sup>[1]</sup> 매크로 / 행위 모델링의 근본적인 목적중의 하나는 회로를 단순히 모델링하여 시뮬레이션 시간을 단축시키는 것이지만, 모델의 정확도가 반드시 보장되어야 하며, 시뮬레이션의 수렴 특성 및 안정성 또한 보장되어야 한다.<sup>[2]</sup> 뿐만아니라 개발된 모델을 라이브러리화하여 쉽게 사용할 수 있도록 적용성과 일반성이 유지되어야 한다.

매크로 / 행위 모델링 기법은 기본적으로 다음과 같이 4가지 유형으로 분류될 수 있다.<sup>[3]</sup> i) 회로를 보다 간단한 topology로 줄이는 방법, ii) 입력 및 출력 터미널 특성을 대략적으로 표현하는 방법, iii) 부회로(sub-circuit) 응답을 커브 혹은 테이블화하여 나타내는 방법, iv) 수학적 표현을 이용한 함수 기술 등이 그것이다. 이와 같은 4가지 유형의 기법은 혼합되어 사용될 수 있으며, 적용하고자하는 회로 시뮬레이터의 지원 능력에 따라 제한적일 수 있다.

국외의 상용 소프트웨어 회사인 Analogy의 SABER<sup>[4]</sup>라는 툴은 실제의 아날로그 회로로부터 매크로 모델을 자동 생성하여 주는 기능을 갖고 있으며, 이러한 기능을 위해 기본적으로 AHDL (Analog Hardware Description Language)를 제공하고 있다. 그러나, 이와 같이 AHDL을 사용하는 툴들은 SPICE 계열의 입력문법을 사용하는 범용 회로 시뮬레이터들과 호환성에 제약을 갖는 단점을 갖고 있다. 반면에 SPICE의 built-in 소자를 사용하는 것은 실제 회로의 스펙과 매크로 모델의 연관성을 유지하기에 용이한 면이 있다. SABER와 유사한 기능을 갖는 툴로서 Anacad사의 ELDO 등이 상용되고 있는데 이들의 공통된 특징은 사용자가 매크로 모델을 편리하게 사용할 수 있도록 하는 환경을 제공하는 것이다.

본 논문의 BEST는 매크로 / 행위 모델 개발에 필요한 기능을 하나의 시스템으로 제공하기 위한 것으로서, 행위 모델의 각 파라미터들을 트랜지스터 레벨 회로의 특성이 잘 반영되도록 최적화해 주는 것과 매크로 모델의 방정식을 스펙에 따라 계산해 주는 것이 주기능으로 제공된다. 특히, 매크로 모델 방정식 계

산 기능은 이전의 매크로 모델 내장 방식<sup>[5]</sup>과는 달리 외부접근 방식으로 이루어져있기 때문에 임의의 매크로 모델에 대해서 사용이 가능하다. 이 밖에도, 논리 회로에 사용되는 MOSFET 트랜지스터를 하나의 스위치로 특성화(characterization)하여 보다 빠른 시뮬레이션이 가능하도록 하는 기능과, 스위치 및 매크로 / 행위 모델 라이브러리를 생성하여 트랜지스터 레벨로 구성된 네트리스트상에 대치시켜주는 기능이 있다. 이와 같은 기능들은 메뉴 선택방식의 그래픽 사용자 인터페이스<sup>[6]</sup>를 통해 이용함으로써 매크로 / 행위 모델의 개발 및 사용의 편리성을 높였으며, 매크로 / 행위 / 스위치 / 트랜지스터 레벨의 회로들이 혼재된 멀티레벨 시뮬레이션이 가능하므로 정확도와 수행속도간의 trade-off를 사용자가 결정할 수 있도록 되었다.

BEST는 위와 같은 기능들을 SPICE 계열의 입력문법을 사용하는 회로 시뮬레이터에 적용함으로써 앞에서 기술된 상용 툴들보다 소프트웨어 이식성과 실제회로의 연관성을 향상시켰으며, 매크로 / 행위 모델의 개발 및 사용에 편리한 환경을 통합된 시스템으로 제공하여 현재 사용하고 있는 시뮬레이터의 기능을 보다 효과적으로 이용할 수 있도록 하였다.

본 논문의 II장에서는 BEST를 이루고 있는 각 모듈 및 시스템 구성에 대한 전반적인 설명을 하였고, III장에서는 제공되는 기능과 구현된 알고리즘에 대해 기술하였으며, IV장에 실험결과를 나타냄으로써 BEST의 성능을 알아볼 수 있도록 하였다. 마지막으로 V장의 결론 및 향후계획을 통해 본 소프트웨어의 발전가능성을 언급하였다.

## II. BEST의 구성

BEST는 그림 1에서와 같이 크게 5가지 모듈로 구성되어 있다.

### 1. Macro 모듈

트랜지스터 레벨에서 구현된 부회로와 종속전원등 기타 회로 시뮬레이터<sup>[7]</sup>에서 제공되는 element들로 구성된 매크로 / 행위 모델을 입력으로 받아들여 트랜지스터 레벨 부회로의 동작특성을 매크로 / 행위 모델이 정확히 묘사하도록 파라미터들을 최적화한다. 또한, 트랜지스터 레벨 회로의 출력과 매크로 / 행위 모델의 출력을 자동적으로 비교해주는 기능을 한다.

### 2. Switch 모듈

크기(width 및 length)가 기술된 MOSFET 리스

트를 받아들여 주어진 조건(모델 파라미터, 온도, 동작 바이어스)하에서 각각의 트랜지스터를 가변선형저항(PWL resistor)과 캐패시터로 구성되는 스위치로 특성화한다.

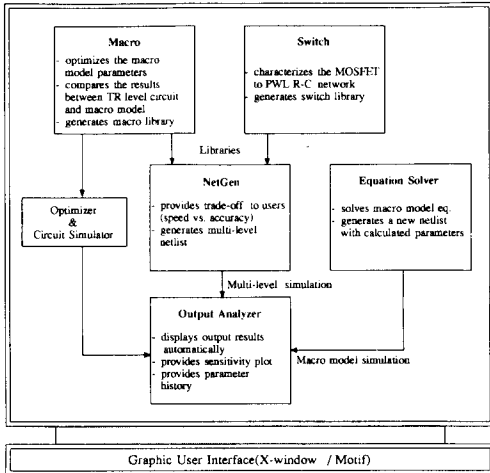


그림 1. BEST 구성도  
Fig. 1. Configuration of BEST.

3. NetGen 모듈

트랜지스터 레벨의 네트리스트를 받아들여 명시된 매크로 / 행위 라이브러리와 스위치 라이브러리에 등록된 각 부회로를 치환하여 멀티레벨 네트리스트를 발생시키는 기능을 한다.

4. Equation Solver 모듈

주어진 스펙에 대해 매크로 모델 방정식을 풀어 파라미터값들을 계산해내며 이 값을 이용하여 새로운 네트리스트를 발생시켜 준다.

5. Output Analyzer 모듈

최적화 및 비교 시뮬레이션이 종료되었을 경우 자동적으로 결과 커브를 발생시키며, 최적화를 위한 sensitivity plot, 파라미터 / 에러 history plot 등을 제공한다.

III. BEST의 기능

1. 매크로 / 행위 모델의 파라미터 최적화 기능

이 기능은 Macro 모듈에서 수행되는 것으로서 트랜지스터 레벨 회로에 대한 매크로 또는 행위 모델의 파라미터값을 최적화시킨다. 최적화를 위한 입력으로

는 트랜지스터 레벨의 부회로(또는 트랜지스터 레벨 부회로의 결과), prototype의 매크로 / 행위 모델, 입력 파형, measure statement, 바이어스 조건 및 온도 조건등이 주어지는데 이 같은 최적화를 위해 다음과 같은 방식들이 지원된다. i) Delay 최적화 방식 : 트랜지스터 레벨 회로의 시뮬레이션 결과에서 특정 지연시간을 target으로 하여 매크로 / 행위 모델의 파라미터들을 최적화함. ii) Curve-fit 최적화 방식 : 트랜지스터 레벨 회로의 시뮬레이션 결과 커브를 target으로 하여 파라미터들을 최적화함. iii) Function 최적화 방식 : 트랜지스터 레벨 회로의 시뮬레이션 결과의 RMS(Root Mean Square), average 값 등을 target으로 하여 파라미터들을 최적화함. 이와 같은 최적화 기능은 메뉴 패널상에서 대화식으로 수행되므로 입력 조건들을 변화해가면서 매크로 / 행위 모델을 개발할 수 있다. 또한, 최적화가 종료되면 결과 파형, 파라미터 / 에러 history plot, sensitivity plot, 최적화 결과 요약 등이 자동적으로 디스플레이되어 사용자가 쉽게 정보를 분석할 수 있도록 하였다. 최적화와 유사한 방식으로 트랜지스터 레벨 회로와 매크로 / 행위 레벨 회로의 시뮬레이션 결과 차이를 분석하여주는 비교 기능도 제공된다. BEST에서 사용된 최적화 알고리즘은 Levenberg-Marquart 방법<sup>19</sup>을 사용하였는데 이 방법에서 최적화의 탐색 방향은 Gauss-Newton 탐색 방식과 Steepest Descent 탐색 방식 사이에 놓여진 점에서 구해진다. 그림 2는 최적화 과정을 나타내는 흐름도이다.

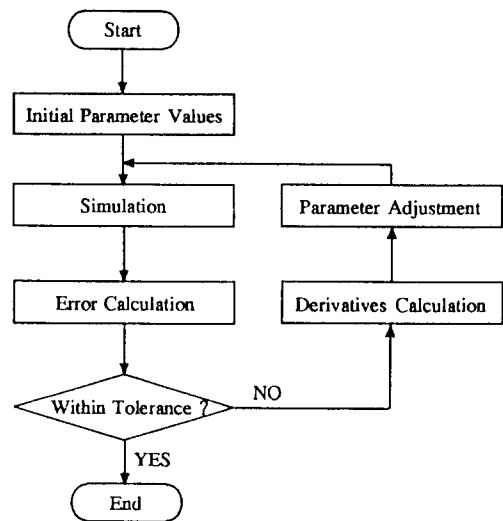


그림 2. 최적화 흐름도  
Fig. 2. Optimization flow.

2. 스위치 특성화 기능

스위치 특성화 기능은 그림 3에서와 같이 비선형 소자인 MOSFET 트랜지스터를 선형 소자인 저항과 캐패시터로 모델링하는 것으로서 시뮬레이션 속도 향상을 목적으로 한다.

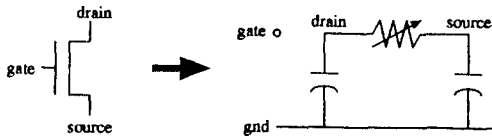


그림 3. 스위치 특성화 모델  
Fig. 3. Switch model.

가변저항(PWL resistor)값은 트랜지스터의 I-V 커브를 이용하여 구하며, 캐패시턴스는 트랜지스터의 출력 시상수 값을 이용하여 정하게 된다. 이 방법을 자세히 살펴보면 다음과 같다.

1) MOSFET의 문턱전압 계산 : 문턱전압은 회로 시뮬레이터의 동작점 해석을 통해 계산되며 모든 작업은 자동으로 이루어진다. 즉, MOSFET의 드레인에는 Vdd 값을, 벌크에는 Vbb 값을 지정하여 시뮬레이션한 후 MOSFET 모델 방정식과 주어진 바이어스에 따라 구해진 문턱전압 값을 추출해내게 된다.

2) 가변저항 모델링 : 그림 3에 나타낸 가변저항은 게이트 노드와 power 노드(NMOS의 경우 접지, PMOS의 경우 Vdd)간의 전압차에 따라 저항값이 변화하는 것으로서 트랜지스터의 게이트 전압 변화에 대한 실효 저항값을 얻기위해 사용되었다. 그러므로 게이트 전압 변화에 따른 드레인-소오스 간의 전류 변화를 나타내는 I-V 커브를 구하고 Ohm의 법칙( $R = I * V$ )을 적용하여 각 게이트 전압에 따른 실효 저항값을 구한다. 여기서 의미하는 실효저항값은 트랜지스터가 turn-on되는 시점에서의 저항값으로 고려한다. 트랜지스터가 turn-on되는 조건은  $(V_g - V_s) > V_{th\_n}$  또는  $(V_g - V_s) < V_{th\_p}$ 로 고려할 수 있으므로 먼저 각 트랜지스터의 threshold 전압을 구해야한다. 또한, 게이트 전압의 변화간격을 결정하여 특성화할 포인트 수를 정한다. 예를 들어 Vdd=5 [V]인 경우에 게이트 전압의 변화 간격을 0.5[V]로 정하면 10포인트의 게이트 전압에서 실효 저항값을 구하게 된다. 그림 4는 I-V 커브상에서 실효저항을 구하는 예를 도식적으로 나타낸 것이다.

3) 캐패시턴스 모델링 : 그림 3에서 캐패시턴스는 드레인 및 소오스 노드와 접지 노드 사이에 존재하게 되며, 앞서 구해진 실효저항과 더불어 트랜지스터의

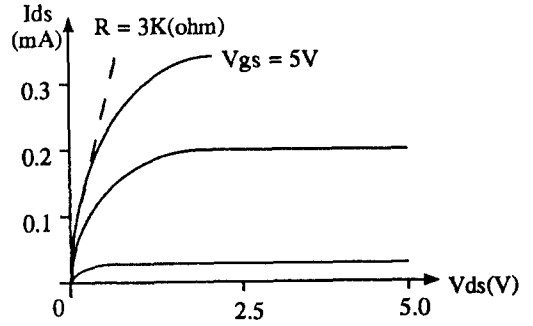


그림 4. I-V 커브에서 실효저항값을 구하는 예  
Fig. 4. Example of extracting the effective resistance values from I-V curves.

지연시간을 나타내게 된다. NMOS 트랜지스터의 스위치 모델링시 캐패시턴스를 구하는 방법은 드레인에 Vdd로 precharge된 전압이 트랜지스터가 turn-on되어 discharge될 때 나타나는 과형의 시상수를 구하고 구해진 시상수값과 앞서 얻은 실효저항을 식 (1)에 대입하여 캐패시턴스값을 계산해내는 것이다.

$$V(t) = V_{dd}(1 - e^{-t/R^C}) \quad (1)$$

여기서 V(t)의 값은 NMOS의 경우 0.63 \* Vdd이고 PMOS의 경우 0.37 \* Vdd이다. 이 때, 전단의 fan-out에 따른 캐패시턴스의 변화는 시뮬레이션 속도를 위해 고려하지 않았으므로 회로의 구성에 따라 모델의 정확도가 다소 차이를 나타낼 수도 있다.

이상에서 구해진 실효저항과 캐패시턴스로 구성된 부회로의 예를 그림 5에 나타내었다.

```

.subckt tni_309u_282u_d g s
gm d s vcr Ppw1(1) g 0
+ 0.00 1.000e+26
+ 0.50 1.000e+26
+ 1.00 1.860e+05
+ 1.50 9.110e+04
+ 2.00 6.331e+04
+ 2.50 6.331e+04
+ 3.00 4.743e+04
cd d 0 1.265f
cs s 0 1.233f
.ends
    
```

```

.subckt tpi_7080u_382u_d g s vdd
op d s vcr Ppw1(1) g s vdd
+ -3.00 4.743e+04
+ -2.50 5.234e+04
+ -2.00 6.331e+04
+ -1.50 9.110e+04
+ -1.00 1.860e+05
+ -0.50 1.000e+26
+ -0.00 1.000e+26
cd d 0 1.832f
cs s 0 1.733f
.ends
    
```

(a) (b)

그림 5. PWL element를 이용하여 모델링된 스위치 부회로 (a) NMOS 스위치 모델 (b) PMOS 스위치 모델

Fig. 5. The subcircuits of switch model which consists of PWL element (a) NMOS switch model (b) PMOS switch model.

그러나 그림 5와 같이 부회로를 구성한 후 트랜지스터와 대처하여 시물레이션할 경우 하나의 노드에 여러개의 캐패시터가 병렬로 연결되는 경우가 발생하게 되어 element 수가 증가되므로 시물레이션의 효율이 떨어지게 된다. 이와 같은 문제점을 해결하기 위해 부회로의 내용중 캐패시터(Cd 및 Cs)를 삭제한 후 라이브러리를 구성하며 이 삭제된 캐패시터는 또 다른 라이브러리에 등록된다. 분리된 캐패시터는 멀티레벨 넷리스트 발생시 NetGen 모듈에 의해 다시 각 노드에 연결되는데 이 때, 병렬로 연결된 캐패시터는 하나로 합쳐져서 연결된다.

3. 멀티레벨 넷리스트 발생 기능

이 기능은 시물레이션할 트랜지스터 레벨 넷리스트를 입력으로 받아들인후 BEST에 의해 구축된 매크로 / 행위 모델 및 스위치 모델 라이브러리를 치환하여 멀티레벨 넷리스트를 생성해내는 것이다. 이때 치환할 부회로들은 사용자가 간단히 지정할 수 있으므로 사용자는 시물레이션의 정확도와 수행속도의 trade-off를 결정할 수 있다. 예를 들어 인버터의 경우에 행위 모델이 존재하고 인버터를 구성하는 각 트랜지스터에 대한 스위치 모델 역시 존재할 경우 사용자는 인버터 부회로를 행위 모델로 사용하도록 지정할 수 있을 뿐만아니라 스위치 모델이 사용되도록 지정할 수도 있다. 또한, 정확도가 중요시되는 경우에는 트랜지스터 레벨의 인버터 부회로가 직접 사용될 수도 있다. 멀티레벨 넷리스트가 구성되는 과정은 그림 6에 나타내었다.

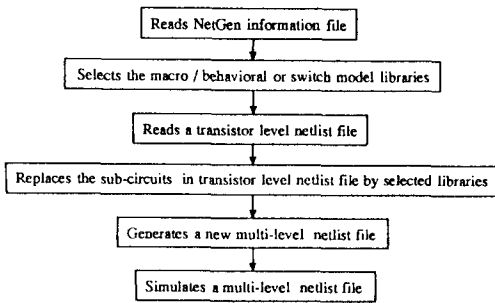


그림 6. 멀티레벨 넷리스트 구성 과정  
Fig. 6. The procedure to generate a multi-level netlist.

4. 매크로 모델 방정식 계산(equation solver) 기능  
이 기능은 매크로 모델 개발시 빈번히 수행되는 방정식 계산을 자동으로 쉽게 수행할 수 있도록 하는 것

이다. 기존의 방정식 자동 계산 방식은 특정한 매크로 모델의 방정식을 프로그램 내부에 저장(hard coding)하여 놓고 필요한 모델을 선택하여 수행하는 방식이었다.<sup>10</sup> 그러나, 매크로 모델의 종류와 그 스펙의 종류는 매우 다양하기 때문에 사용에 제약이 있었다. BEST에서는 이러한 문제점을 해결하기 위해 방정식 및 스펙을 외부에서 읽어들이어 계산하는 방식(soft coding)을 채택하였다.<sup>11</sup> 이 때 사용자가 수행해야 하는 방정식 타이핑 작업은 외부에 라이브러리를 구함으로써 해결할 수 있다. 또한 사용자가 매크로 모델 회로를 기술하였을 경우에는 계산된 파라미터값을 이 회로의 넷리스트 상에서 치환하여 새로운 매크로 모델 넷리스트로 발생해주는 기능이 있다. 그림 7에 매크로 모델 방정식의 계산과정을, 표 1에는 방정식에 사용될 수 있는 수학적 함수를 나타내었다.

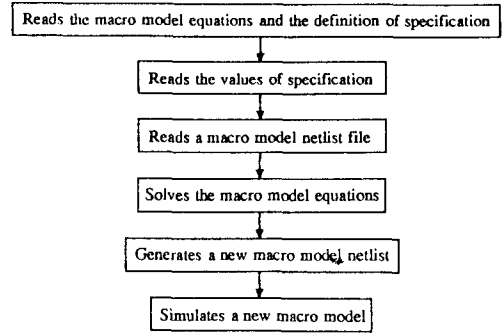


그림 7. 매크로모델 방정식의 계산 과정  
Fig. 7. The procedure to calculate the equations of macromodel.

표 1. 매크로 모델 방정식에 사용가능한 수학적 함수 표현

Table 1. The mathematical function that can be used in macromodel equation.

매크로 모델 방정식에 사용 가능한 수학적 함수 표현	asin(), asinh(), sin(), sinh(), acos(), acosh(), cos(), cosh(), atan(), atanh(), tan(), tanh(), pow(), sqrt(), log(), log10(), exp(), ceil(), fabs()
------------------------------	---

5. 결과 분석 기능

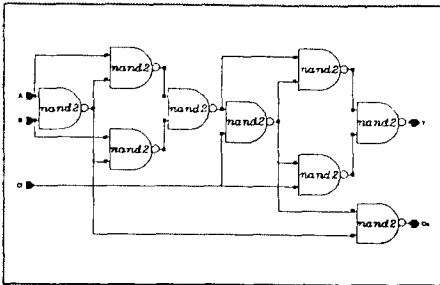
매크로 / 행위 모델 최적화, 멀티레벨 시물레이션, 매크로 모델 시물레이션 등의 결과는 Output Analyzer 모듈에서 용이하게 분석할 수 있다. 최적

화 종료시에는 트랜지스터 레벨의 결과와 매크로 / 행위 모델의 결과 파형이 자동적으로 디스플레이되어 사용자가 쉽게 분석할 수 있으며 이 결과들은 화일로 저장되어 이후에 쉽게 다시 분석할 수 있다. 또한, 최적화의 결과를 보여주는 결과 요약 패널과 최적화 파라미터들이 출력에 미치는 기여도를 나타내는 sensitivity plot 등이 제공되며, 최적화 진행에 따른 파라미터의 변화를 기록하는 파라미터 history 및 에러 history plot 등이 제공된다.

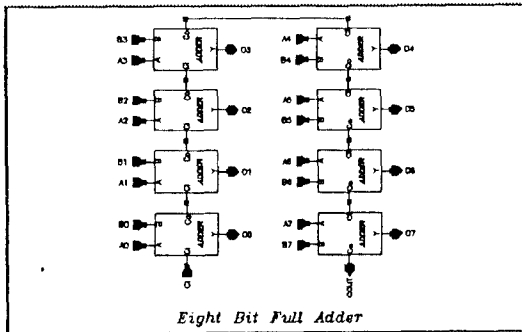
#### IV. 실험 결과 및 고찰

##### 1. 디지털 회로의 적용

앞에서 설명된 BEST의 기능중 멀티레벨 시뮬레이션 기능을 테스트하기 위해 CMOS 논리 회로인 8-bit adder 회로를 대상으로 선정하였다. 테스트의 목적은 스위치 및 매크로 / 행위 모델을 시뮬레이션에 적용할 경우에 얻어지는 수행속도 개선 정도 및 출력 파형의 정확도를 알아보고자 함이다.



(a)



(b)

그림 8. 테스트 회로(8-bit adder)(a) 1-bit adder (b) 8-bit adder

Fig. 8. Test circuit(8-bit adder)(a) 1-bit adder (b) 8-bit adder.

사용된 adder 회로는 1-bit adder를 기본으로 구성되어 있으며 사용된 MOSFET의 크기는 NMOS의 경우 너비/길이 = 5um/1.25um, PMOS의 경우 10um/1.25um이다. 그림 8에 이 회로에 대한 스키메틱을 나타내었다.

멀티레벨 시뮬레이션을 하기위해 먼저 1-bit adder에 대한 행위 모델을 제작하였다. 이 모델의 논리 함수는 회로 시뮬레이터에서 제공되는 polynomial 함수를 이용하여 구성하였으며, 지연시간 및 파형의 기울기를 맞추기 위하여 delay element, 저항 및 캐패시터를 추가하였다. 그림 9에 1-bit adder 행위 모델의 네트리스트를 나타내었는데 여기서 td1, td2, td3, resout, rescout, capout, capcout 등의 파라미터는 BEST를 이용하여 최적화 하였다. 그 다음 트랜지스터 레벨의 adder 회로에 기술된 MOSFET들에 대해 스위치 특성화를 수행하여 라이브러리를 구축하였다. 여기서 사용된 모델은 다양한 fan-out을 고려하지 않은 모델이기 때문에 복잡한 회로에서 사용할 경우 정확도가 떨어질 수도 있다.

```
.subckt onebit in1 in2 cin out cout
e1 tout 0 POLY(3) tin1 0 tin2 0 tcin 0
+ 0 1 1 1 0 -0.4 -0.4 0 0 0 0 0 0.16
e2 tcout 0 POLY(3) tin1 0 tin2 0 tcin 0
+ 0 0 0 0 0 0.2 0.2 0 0.2 0 0 0 0 0 -0.8
ed1 tin1 0 DELAY in1 0 td=td1
ed2 tin2 0 DELAY in2 0 td=td2
ed3 tcin 0 DELAY cin 0 td=td3
rout tout out resout
rcout tcout cout rescout
cout out 0 capout
ccout cout 0 capcout
.ends
```

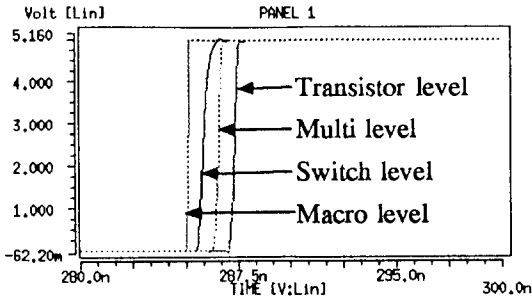
그림 9. 1-bit adder의 행위 모델

Fig. 9. Behavioral model for 1-bit adder.

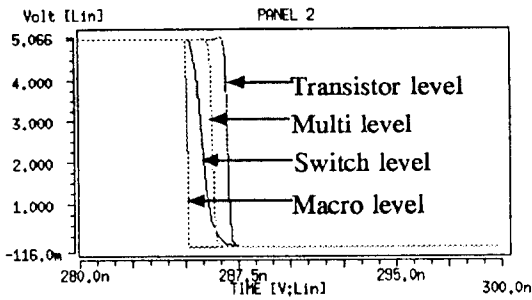
위와 같은 작업을 수행한 후 행위 레벨, 스위치 레벨, 트랜지스터 레벨 및 멀티레벨로 각각 시뮬레이션하여 결과를 비교하였다. 앞에서 언급한 각각의 시뮬레이션을 수행한 후 8-bit adder의 마지막 단의 sum과 carry-out에 대한 파형을 Output Analyzer를 이용하여 그림 10에 비교하였다.

그림 10에서 보듯이 각 레벨의 시뮬레이션 결과가 트랜지스터 레벨 시뮬레이션 결과와 잘 맞는 것을 알 수 있다. 표 2에 테스트 결과를 시뮬레이션 시간과 오차 측면에서 나타내었다.

표 2에서 보면 각각의 레벨에 따라 시뮬레이션 속도와 정확도간에 trade-off가 나타나는 것을 알 수 있다. 특히 행위 레벨 시뮬레이션의 경우에는 18.6배의 속도 향상을 나타내었으며, 멀티레벨의 경우 속도와 정확성에서 모두 양호한 결과를 나타내었다.



(a)



(b)

그림 10. 트랜지스터, 행위, 스위치, 멀티 레벨의 시뮬레이션 결과 비교 (a) 8-bit adder 마지막 단의 sum 파형 비교 (b) 8-bit adder 마지막 단의 carry-out 파형 비교

Fig. 10. The comparison of the results from transistor, behavioral, switch and multi-level simulation

(a) The comparison of the sum curves from last stage of 8-bit adder (b) The comparison of the carry-out curves from last stage of 8-bit adder.

표 2. 각 레벨별 시뮬레이션 결과 비교

Table 2. The comparison of the result from the simulation on each level.

시뮬레이션 레벨	수행 시간(sec)	지연시간 차이(msec)	
		sum	carry-out
트랜지스터 레벨	528.55	refnce	reference
스위치 레벨	345.10	1.82	1.35
행위 레벨	28.48	2.1	1.52
멀티 레벨	361.43	0.95	0.94

2. 아날로그 회로의 적용

BFEST의 매크로 / 행위 모델 최적화 기능을 테스트하기 위한 회로로서 아날로그 회로인 op-amp를 선정하였다.

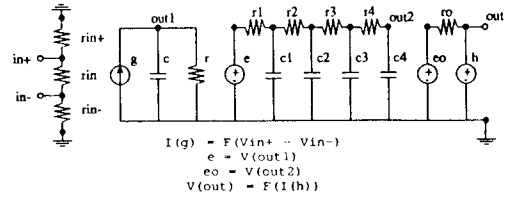
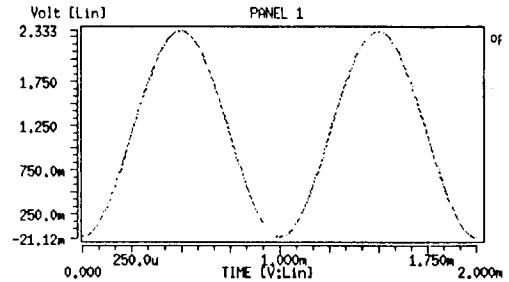
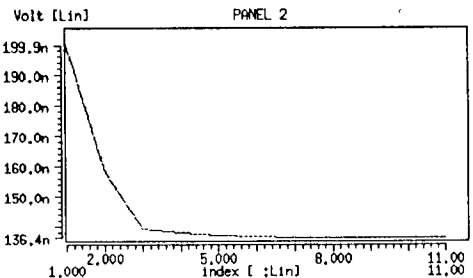


그림 11. Op-amp 매크로 모델 회로도

Fig. 11. Op-amp macromodel.



(a)



(b)

그림 12. Op-amp 매크로 모델의 최적화 결과

(a) Op-amp 매크로 모델의 출력 결과와 커브 target와의 비교 (b) 최적화 파라미터 c의 변화 과정

Fig. 12. The result from the optimization for op-amp macromodel (a) The comparison between the output of op-amp macromodel and target curve (b) The history of the optimized parameter c.

본 검증에 사용된 op-amp 마이크로 모델은 그림 10과 같으며, 이것은 입력단, pole을 나타내는 회로 및 출력제한 회로 등으로 구성된다.

그림 11에서 입력단의 파라미터인 c는 op-amp의 transient 해석시 출력의 magnitude에 영향을 미치는 것으로서 이것을 BEST의 마이크로 / 행위 모델 파라미터 최적화 기능을 통하여 최적화 하였다. 그 결과 그림 12-(a)와 같이 target에 overlap된 출력 파형을 얻을 수 있었으며, 최적화 파라미터 c의 변화 과정은 그림 12-(b)와 같이 나타났다.

3. 그래픽 사용자 인터페이스

앞에서 수행된 모든 작업들은 그림 13, 14에 나타난 메뉴 선택 방식의 그래픽 사용자 인터페이스를 통하여 수행되었다. 그림 13은 마이크로 / 행위 모델의 최적화 환경을 나타내며, 그림 14는 Switch, NetGen 및 Equation Solver 모듈의 수행 환경을 나타낸다. 이런 인터페이스를 통해 사용자는 편리하게 필요사항을 입력하고 결과분석도 그래픽적으로 출력되어 대화형으로 시스템을 사용할 수 있다.

V. 결론 및 향후 계획

앞에서 언급한 바와 같이 시뮬레이션 수행 속도 개선을 위하여 마이크로 / 행위 모델링 기법이 널리 연구되고 있으며, 본 논문에서 기술된 BEST는 이러한 모델링 기법을 지원하기 위해 개발되었다. BEST를 이용하여 실험 회로에 대해 적용해본 결과 마이크로 / 행위 모델의 파라미터 최적화를 통해 정확한 모델링을 수행할 수 있었으며 여러가지 레벨에 대해 시뮬레이션 속도와 정확도간의 trade-off를 사용자가 자유롭게 선택할 수 있음을 알 수 있었다. 마이크로 / 행위 모델은 대상회로의 크기와 모델의 복잡도에 따라 시뮬레이션 속도 향상이 크게 변화할 것이라고 생각되는데 이러한 모델을 잘 개발하려면 다음과 같은 사항이 준비되어야 할 것이다. 첫째, 설계자의 마이크로 / 행위 모델링 능력이 준비되어야 한다. 즉, 회로 시뮬레이터에서 제공되는 행위 레벨 문법을 숙지하여 트랜지스터 레벨의 회로를 종속전원을 이용한 등가회로 또는 수학적 표현을 이용한 기능 모델 등으로 모델링할 수 있어야 한다. 둘째, 시뮬레이터의 행위 레벨 문법 자체의 기능이 보완되어야 한다. 기존의 회로 시

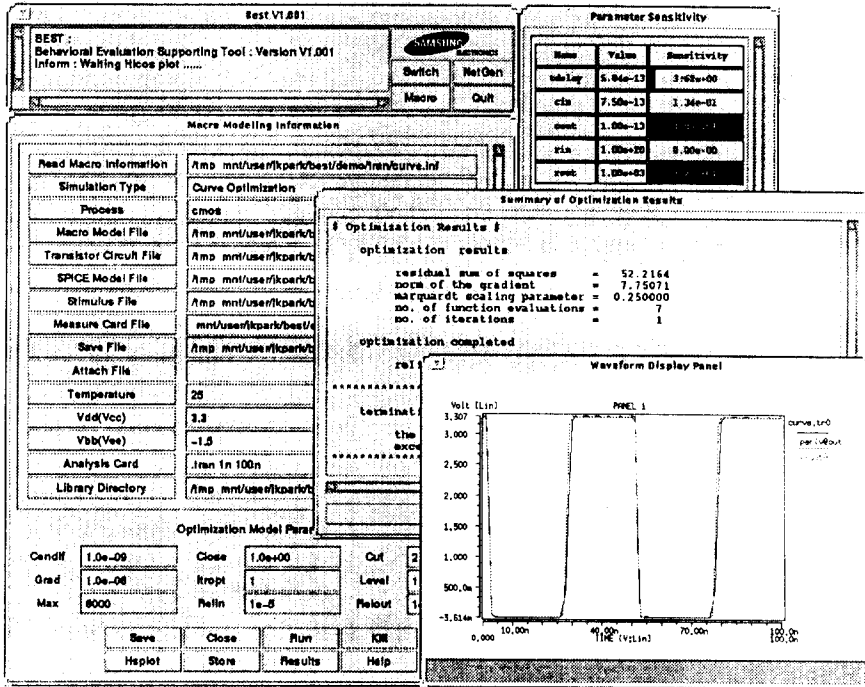


그림 13. Macro 모듈의 수행환경  
Fig. 13. Graphic user interface for Macro module.



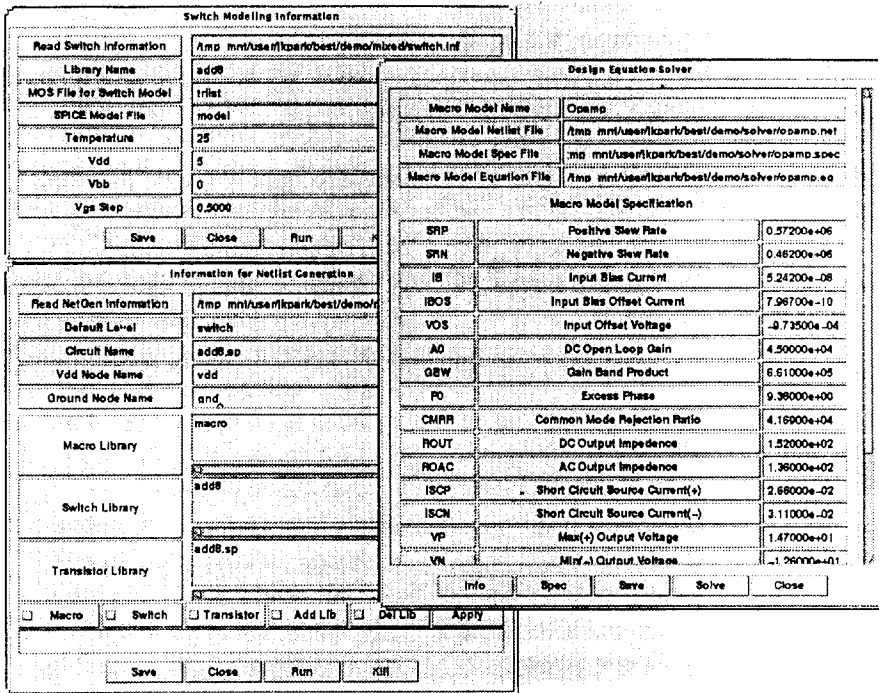


그림 14. Switch, NetGen 및 Equation Solver 모듈의 수행환경  
 Fig. 14. Graphic user interface for Switch, NetGen and Equation Solver module.

물레이터 문법내에 IF-THEN-ELSE와 같은 조건문 이 제공되는 것이 바람직하며 PWL 및 polynomial 함수의 차수 제약이 보완 되어야할 것이다.

향후 BEST에 보강되어야할 항목들은 다음과 같다. i) 매크로 모델의 시뮬레이션 특성 검사 기능 : 개발된 매크로 모델이 시뮬레이션에 적용될 때 연속성, numerical overflow 등 시뮬레이션 수행에 치명적인 영향을 미칠 소지가 있는 지를 검사하는 기능. ii) 행위레벨 언어 제작 : BEST에 자체적인 행위레벨 언어(예를 들어 C 언어 또는 VHDL style)를 내장하여 사용자가 이 언어로 회로의 행위를 기술하면 자동으로 회로 시뮬레이터의 문법에 맞게 변환해주는 기능이다.

앞으로 회로의 크기가 기하급수적으로 증가하게 되면 매크로 / 행위 모델 개발이 더욱 중요한 역할을 할 것이라고 생각되며 BEST 또한 이와 같은 문제에 대비하는 방향으로 개선해 나갈 것이다.

參考文獻

[ 1 ] 아나로그 모델링, 삼성첨단기술연구소, 1992  
 [ 2 ] Yun-Cheng Ju and Resve A. Saleh,

"Consistency Checking and Optimization of Macromodels", IEEE Trans. on CAD, vol.10, no.8, Aug. 1991.  
 [ 3 ] A. E. Ruehli, R. B. Rabbat and H. Y. Hsieh, "Macromodeling-An approach for analysing large-scale circuits", Computer-Aided Design, vol.10, no.2, pp.121-130, Mar. 1978.  
 [ 4 ] SABER Technical Note, Analogy.  
 [ 5 ] J. Alvin Connelly and Pyung Choi, Macromodeling with SPICE, Prentice-Hall, 1992.  
 [ 6 ] OSF / Motif Programmer's Guide, Prentice-Hall, 1990.  
 [ 7 ] HSPICE User's Manual, Meta Software, 1991.  
 [ 8 ] Lynne M. Brocco, Steven P. McCormick and Jonathan Allen, "Macromodeling CMOS Circuits for Timing Simulation", IEEE Trans. on CAD, vol.7, no.12, Dec. 1988.  
 [ 9 ] D. E. Ward and K. Doganis, "Opti-

mized Extraction of MOS Model Parameters". IEEE Trans. on CAD, vol.CAD-1, no.4, Oct. 1982.

[10] A. E. Ruehli, Circuit Analysis, Simulation and Design 2, North-Holland,

1987.

[11] Axel T. Schreiner and H. George Friedman Jr., Introduction to Compiler Construction with UNIX, Prentice-Hall, 1985.

著 者 紹 介



朴 璉 圭(正會員)  
1990年 2月 연세대학교 전기공학과 졸업(학사). 1990年 ~ 현재 삼성전자 CAE 주임연구원. 주관심 분야는 회로 시뮬레이션, 마크로 모델링, MOSFET 모델링, VLSI 회로 설계 자동화 등임.

鄭 義 英(正會員)  
1988年 2月 고려대학교 전자공학과 졸업(학사). 1990年 2月 동 대학원 졸업(석사). 1990年 ~ 현재 삼성전자 CAE 주임연구원. 주관심 분야는 논리 시뮬레이션, 논리 합성, 테스트, VHDL 모델링 등임.



金 敬 昊(正會員)  
1984年 2月 연세대학교 전자공학과 졸업(학사). 1987年 2月 한국과학기술원 전기 및 전자공학과(공학석사). 1991年 2月 한국과학기술원 전기 및 전자공학과(공학박사). 1983年 ~ 현재 삼성전자 ASIC 및 CAE 선임 연구원. 주관심 분야는 VLSI CAD 및 TCAD 등임.