

심볼릭 시뮬레이션 기법을 이용한 RTL 스캔 설계 법칙 검사기

이종훈⁰ 민형복
(주)디엠티, 성균관대학교
jhlee@dmts.com, min@stella.skku.ac.kr

RTL Design Scan Rule Checker Based On Symbolic Simulation

Jong-Hoon Lee⁰ Hyoung-Bok Min
DMT Co., Sung Kyun Kwan Univ.

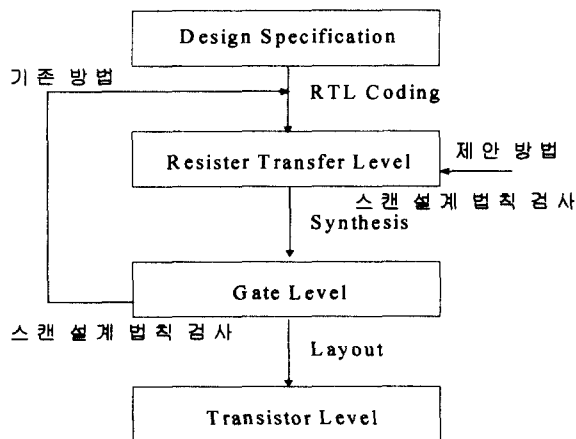
요 약

전통적으로 스캔 설계 법칙 검사는 게이트 레벨에서 수행되었다. 그러나 RTL 설계와 합성 도구의 사용이 일반화되면서 게이트 레벨 회로의 검사는 합성 단계에서의 최적화와 스캔 설계 법칙 위배를 정정한 후의 최적화가 필요하여 많은 시간이 소요된다.

RTL에서의 스캔 설계 법칙 검사는 이러한 문제를 해결할 수 있으며, 이것이 본 논문의 주제이다. 본 논문에서는 스캔 설계 법칙의 위배를 RTL 설계에서 검사할 수 있는 기법을 제안한다. 이 기법은 효과적인 설계 과정에 의해 설계 시간을 단축할 수 있을 것이다.

※ 본 연구는 2000년도 한국과학재단 목적기초 연구지원에 의한 결과임 (과제번호 : 2000-1-30200-002-3).

1. 서 론



(그림 1) ASIC 설계 단계

본 논문에서는 디지털 회로 설계에 널리 사용되고 있는 Verilog HDL로 설계된 RTL 디자인들에 스캔 설계 법칙을 적용하여 검증하는 방법을 제시하고자 한다. (그림 1)에서와 같이 기존의 방법은 합성을 통해 생성된 게이트 단계의 디자인에 스캔 설계 법칙을 적용하여 설계 오류가 발견되면 HDL 코드

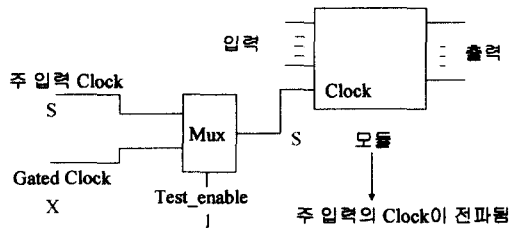
에 수정을 하고 다시 합성을 통해 게이트 단계의 설계를 얻어 낸 후에 오류 정정을 확인해야만 했다. 본 논문에 제안하는 방법은 이렇게 피드백 루프로 수행되던 스캔 설계 법칙 검사를 RTL에서 직접 수행함으로써 합성 전에 디자인의 오류를 찾아 내어 설계의 변경을 좀 더 빠른 시간 안에 가능하도록 할 수 있다. 본 논문에서 구현된 RDRC(RTL Design Rule Checker)에서는 조건문의 기술이 완벽하지 않았을 때 설계자가 의도하지 않았던 래치가 생성되는가를 검사하고, HDL의 문법과 구문을 검사함으로써 쉽게 발견 할 수 없는 Gated Clock과 Gated Reset이 존재하는지를 검사한다. 특히 Gated Clock이나 Gated Reset은 RTL 코드의 분석이나 하위 모듈들간의 연결 정보만으로는 발견해 내기가 불가능하다. 따라서 본 논문에서 구현된 RDRC에서는 기존의 게이트 레벨에서 적용되어 오던 Symbolic Simulation 기법^[3]을 RTL 코드에 적용하여 Gated Clock과 Gated Reset을 효율적으로 찾아내는 방법을 제시하고자 한다.

2. Symbolic Simulation

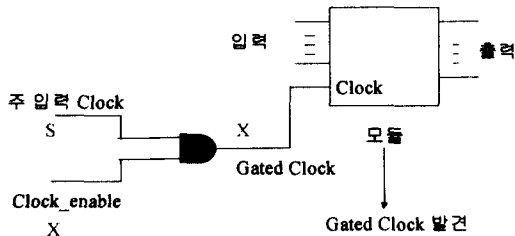
Symbolic Simulation이란 주 입력 단의 특정 입력 값에 Symbol을 인가하고 전체 회로에 대한 Simulation을 통해 그 Symbol이 특정 하위 모듈의 입력 단까지 전파되어 지는지를 검사하여 Gated Clock을 찾아내는 방법을 말한다.^[3]

이러한 Symbolic Simulation에는 0, 1, X, S, \bar{S} 이상의 다섯 가지 논리 값이 사용된다.

(그림 2)에서 최상위 모듈의 Clock 입력에 S를 인가하고 Test_enable과 같은 특정 값을 1로 고정하고, 다른 모든 입력 값에는 X를 인가한 다음 Symbolic Simulation을 수행한 후 하위 모듈의 Clock 입력의 값을 검사해 보면 S 값이 관찰되어짐을 볼 수 있을 것이다.



(그림 2) 심볼릭 시뮬레이션을 이용한 Clock 전파 확인
그러나 (그림 3)과 같은 회로에서는 하위 모듈의 Clock이 X로 나타나게 된다.



(그림 3) 심볼릭 시뮬레이션을 이용한 Gated Clock 발견
이러한 방법에 의해 검사 하고자 하는 하위 모듈의 Clock 입력을 모두 검사하여 Gated Clock이 인가되었는지 주 입력의 Clock이 인가되었는지를 검사 할 수 있게 된다.

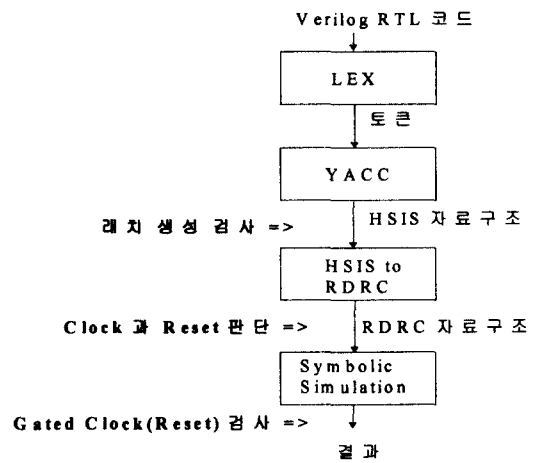
RDRC는 이러한 Symbolic Simulation을 RTL 코드에 적용하여 구현하였다.

3. RDRC (RTL Design Rule Checker)의 구현 및 결과

본 논문에서는 RTL 설계 회로에 대한 스캔 설계 법칙 검사를 수행하는 소프트웨어 RDRC를 구현하였고, 이를 예제에 적용하여 그 결과를 관찰하였다.

3.1. RDRC의 구현

RDRC는 (그림 4)와 같이 동작되며, Verilog Parser는 Berkeley 대학의 HSIS에서 사용된 Verilog Parser를 사용하였다.^[5]



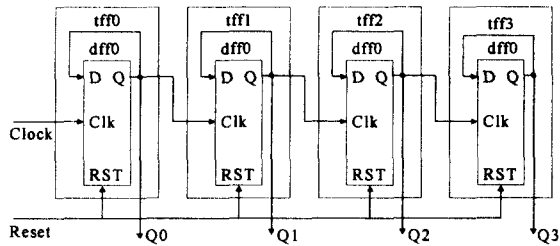
(그림 4) RDRC 전체 흐름도

RTL로 설계된 Verilog 코드를 LEX와 YACC를 이용하여 문법 검사와 레지의 생성 여부를 검사한 후 Symbolic Simulation이 가능한 RDRC 자료 구조로 변경한다. 이 때 Clock과 Reset을 자동으로 찾게된다. 그 다음 Symbolic Simulation을 수행하며 Symbolic Simulation의 과정은 다음과 같다.

- 1) 주 입력 단의 Clock 단에 Symbol S 입력
- 2) 주 입력 단의 다른 입력에 X 값 입력 Test_enable과 같은 특정 입력에는 1 또는 0
- 3) Symbolic Simulation 수행
- 4) 하위 모듈의 Clock 단에 S 값이 전파되었는지를 검사
- 5) S 값이 전파되지 않은 모듈은 Gated Clock 이 인가된 것으로 판단

3.2. RDRC 결과

(그림 5)는 4 bit ripple-carry counter를 모듈별로 계층적 구조를 나타낸 그림이다.



(그림 5) 4-bit 카운터의 계층적 모듈 구조

설계자는 최상위 단계인 ripple_counter 모듈에 test_enable 신호를 추가하여 주고, tff1, tff2, tff3 앞에 Mux를 추가하여 주 입력 단의 Clock이 하위 모듈인 DFF까지 도달할 수 있도록 설계 변경을 해 주어야 한다.

다음은 변경된 설계에 Clock에 S를, test_enable은 1의 값을 인가하여 RDRC를 수행한 결과이다.

```
Port Name : clk Value : S
Port Name : test_enable Value : 1
-- top module name : ripple_counter
Module Name : ripple_counter
  Module Name : T_FF ( tff0 )
    Module Name : D_FF ( dff0 )
      **** Clock Port ****
      clk = S
  Module Name : T_FF ( tff1 )
    Module Name : D_FF ( dff1 )
      **** Clock Port ****
      clk = S
  Module Name : T_FF ( tff2 )
    Module Name : D_FF ( dff2 )
      **** Clock Port ****
      clk = S
  Module Name : T_FF ( tff3 )
    Module Name : D_FF ( dff3 )
      **** Clock Port ****
      clk = S
"RDRC completed !!!"
```

이와 같이 test_enable 입력을 추가하여 테스트 시 주 입력의 Clock이 모든 하위 모듈까지 전파되어 지는 것을 알 수 있다.

4. 결론

본 논문에서는 설계자가 의도하지 않았던 래치의 생성 여부를 설계자에게 알려주어 설계자가 의도하지 않은 래치에 대해서 설계 변경을 하도록 하였고, Gated Clock과 Gated Reset의 검사를 통해 테스트 용이성을 떨어뜨리는 스캔 법칙의 위배를 검사하였다.

이러한 RTL에서의 법칙 검사기는 게이트 레벨에서의 법칙 검사보다 한 단계 위에서 설계 오류를 찾아내어 주므로 설계 시간의 많은 단축을 가져올 수 있다. 본 논문에서 제시한 RDRC는 스캔 법칙의 세 가지 경우에 대해서만 RTL 법칙 검사를 구현하였지만 RTL 설계가 주를 이루는 추세에서 기존의 게이트 레벨에서의 여러 가지 법칙들을 RTL에서 재 정의하고 이를 검사하는 법칙 검사기는 칩 설계에 있어서 많은 이득을 가져올 것이다.

5. 참고 문헌

- [1] M. Abramovici, M. A. Breuer and D. Friedman, "Digital System Testing and Testable Design", Computer Science Press, 1990
- [2] Michael D. Diletti, "Modeling, Synthesis, and Rapid Prototyping with the Verilog HDL", Prentice-Hall, 1999
- [3] Prab Varma, "TDRC - A Symbolic Simulation Based Design For Testability Rules Checker", International Test Conference(ITC) Proceedings, pp. 1055 - 1064, 1990
- [4] Edward B. Pity, Denis Martin, Hi-Keung Tony Ma "A Simulation-Based Protocol-Driven Scan Test Design Rule Checker", International Test Conference(ITC) Proceedings, pp. 999 - 1006, 1994
- [5] Szu-Tsung Cheng, Robert K. Brayton, "Compiling Verilog into Automata", University of California, Berkeley, 1994
- [6] Michael Keating, Pierre Bricaud, "Reuse Methodology Manual", Kluwer Academic Publishers, 1999
- [7] Samir Palnitkar, "Verilog HDL: A Guide to Digital Design and Synthesis", SunSoft Press A Prentice Hall Title, 1996