샘플링을 이용한 설계 오류 시뮬레이션과 시뮬레이션 검증율

Sampling Based Design Error Simulation and Simulation Coverage

이 항규 강 성호
컴퓨터 시스템 연구실
연세대학교

초록

시뮬레이션이 설계 검증에 사용될 때 검증의 척도를 효율적으로 제공하기 위해 설계오류 모델과 통계적인 샘플링 기법을 사용한 시뮬레이션 시스템이 개발되었다. 이는 시뮬레이션 검증율을 제공하고 샘플링 기법을 이용하여 검증율을 빨리 예측할 수 있다. 실험 결과는 이 방법이 효과적임을 증명한다. 이 시스템은 전체 설계시간을 줄이는 효과적인 설계검증 도구로 사용될 수 있다.

1. 서론

디지털 시스템의 크기가 커짐에 따라 이의 설계와 검증은 더욱 어려워지고 있다. 시뮬레이션은 디지털 시스템의 설계를 검증하는데 중요하고도 널리 쓰이는 방법이다. 이는 또한 고장 시뮬레이션이나 테스트 패턴 생성에도 중요한 역할을 한다. 시뮬레이션의 가장 큰 단점은 대상이 크고 복잡해짐에 따라 시간과 비용이 많이 들게 된다는 것이다. 게다가 비용이 너무 커서 거의 사용할 수 없는 전수 시뮬레이션을 제외하고는 시뮬레이션의 완성도를 측정할 수 있는 방법이 없다. 따라서 시뮬레이션 패턴의 일부가 사용되었을 때 얼마만큼 검증되었는지가 궁금하게 된다. 시뮬레이션의 완성도는 어느 정도의 시뮬레이션이 수행되었는지를 결정지어 시뮬레이션을 더 수행해야 할지 아니면 정지할 지를 결정할 수 있게 해준다. 설계 검증의 실제 완성도를 얻기 위하여 주어진 시뮬레이션 패턴에 대한 시뮬레이션 검증율이 필요하다. 이를 해결하기 위한 많은 연구[1,2]가 행해져 왔다.

본 연구에서는 설계 오류 모델링에 기초한 시뮬레이션 검증율을 새롭게 분석하고 이를 토대로 큰 회로를 효과적으로 다룰 수 있는 설계 검증 시스템을 개발한다. 이는 설계 오류 시뮬레이션[1], 설계 오류 시뮬레이션 패턴 생성[3], 그리고 통계적 샘플링 기법[4,5]을 이용하여 주어진 시뮬레이션 패턴에 대해 검증율을 제공한다. 실험 결과는 이 시스템이 설계 검증의 비용을 줄이는데 효과적임을 나타낸다.

2. 시뮬레이션 검증율의 분석

주입력의 수를 n이라 하고 주출력의 수를 u라 하자. 그러면 0과 1만이 입력으로 들어온다고 가정하면 조합회로의 경우 $M=2^n$의 가능한 시뮬레이션 패턴과 $N=2^M$의 가능한 함수가 존재한다. 시뮬레이션 패턴을 $p_1, p_2, \ldots p_M$이라 하고 함수를 $f_1, f_2, \ldots f_N$이라 하자. 또한, 입력을 $j_1, j_2, \ldots, j_n$, 그리고 출력을 $o_1, o_2, \ldots, o_u$라 하자. 만약 어떤 함수의 설계가 주어졌을 때 N중에서 다른 가능성은 모두 설계 오류가 된다. 회로에 어떤 오류가 있더라도 회로의 출력이 원하는 동작을 하게 하면 이는 더 이상 설계 오류가 아니다. 어떤 함수 f_j에 설계 오류 a가 있을 때 이 설계 오류가 있는 함수

를 f_a라 하면 설계 오류 a는 $f_j(p_i) \oplus f_a(p_i)=1$을 만족시키는 입력 p_i에 의해 검출될 수 있다.

2.1 시뮬레이션 패턴에 기초한 검증율

시뮬레이션 패턴에 기초한 검증율(SCP)[8]은 시뮬레이션 패턴을 이용하여 다음과 같이 정의 된다.

SCP = 시뮬레이션에 사용된 패턴의 수 / 가능한 패턴의 수 $\times 100[\%]$

이는 각 설계 오류의 발생 확률과 각 시뮬레이션 패턴의 효율성이 같다는 가정에 기초한 것이다. 이는 시뮬레이션 검증율이 시뮬레이션에 사용된 패턴의 수에 비례하게 되지만 우리의 경험적 직관은 패턴이 다른 가중치를 지님을 알 수 있다.

2.2 이론적 분석

설계 오류 모델이 고려될 때 시뮬레이션 검증율은 다음과 같이 나타내어 진다.

검출된 오류의 수 / 전체 오류의 수 $\times 100[\%]$

순차회로의 경우 내부 상태에 따라 주어진 패턴에 대해 출력이 한 개 이상의 값을 가질 수 있다. 패턴 p_i에 의해 검출되는 오류의 집합을 E_i라하고 E_i의 크기를 $||E_i||$하고 출력 u에서 검출된 오류의 수를 E_u라 하자. 그러면 s개의 패턴이 가해질 때 전체 검출된 오류의 수는 아래와 같다.

$$E = \bigcup_{i=1}^{u} [\bigcup_{j=1}^{s} E_j^i]$$

회로 전체의 오류의 수가 각 출력에서의 오류의 수의 합이라 가정하자. 출력 o_i에 영향을 주는 입력의 수를 q(i)라 하고 출력 o_i에 가해지는 패턴중 같은 패턴의 수를 차수라하고 이를 d_i로 나타내자. 차수란 출력 o_j에 영향을 주는 입력에 같은 값이 들어가는 경우를 표현한다. 출력 o_i에 영향을 주는 입력을 $j_1, j_2, \ldots, j_{q(i)}$라 하자. 첫째 패턴이 $j_1=j_2= \ldots = j_{q(i)}=1$, $j_{q(i+1)}= \ldots =j_n=1$이고 둘째 패턴이 $j_1=j_2= \ldots = j_{q(i)}=1$, $j_{q(i+1)}= \ldots =j_n=0$이면 두 패턴은 o_i에 같은 출력을 내보낸다. 따라서 첫째 패턴이 인가된 후 둘째 패턴이 인가 되면 둘째 패턴으로는 설계 오류가 하나도 검출되지 않는다. 따라서 $||E_i|| = \sum_{(1 \le k \le N,\ k \ne j_s,\ 1 \le t \le r)} \prod_{s=1}^{r} [f_k(p_i) \oplus f_t(p_i)]$ 로 주어진다.

X 값의 수를 x라 하면 오류의 수는 $N - 2^x$이다. $||E||$를 계산하기 위해 X 패턴 상수, c를 고려하는데 이는 출력을 X로 만드는 가해진 패턴의 수이다. 다시 말하면 $c= \sum_{I=1}^{1} (Z(i))$이고 여기서 Z(i)는 $f_j(p_i)$ = X이면 1이고 아니면 0이다. $2^{q(i)}$를 Q(i)라 하면 회로내의 오류의 수는 $\sum_{i=1}^{u} (2^{Q(i)} - r_i)$이다. $Y(a,b) = 2^{a-1} + 2^{a-2} + \ldots + 2^{a-b}$라 하면 s개의 패턴을 이용해 출력 o_j에서 검출되는 오류의 수가 $Y(2^{q(i)}, s-d_i-c_i)$이므로 s개의 패턴으로 검출되는 전체 오류의 수는 $|| E || = \sum_{i=1}^{u} Y(2^{q(i)}, s-d_i-c_i)$ 이고 여기서 나타낸다.

따라서 임의의 회로에 대한 일반적인 시뮬레이션 검증율은 아래와 같다.

$$\sum_{i=1}^{u} [Q(2^{q(i)}, s-d_i-c_i)]/[2^{Q(i)} - r_i] \times 100[\%]$$

단일 출력을 갖는 조합회로에서 입력의 수가 n이고 모든 입력이 출력에 영향을 미친다면 시뮬레이션 검증율은 $[Q(N, 1)]/[M-1] \times 100[\%]$로 표현된다.

2.3 설계 오류 모델에 기초한 시뮬레이션 검증율

완벽한 즉 100%의 시뮬레이션 검증율을 얻기 위해서는 가능한 모든 패턴을 시뮬레이션에 사용하여야 한다. 그러나 위의 이론적인 분석은 시뮬레이션 패턴의 수가 시뮬레이션 검증율에 비례하지 않음을 나타낸다. 이 분석은 설계 오류의 수에 기초한 검증율이 SCP보다 더 정확함을 나타낸다. 이론적 분석은 효율적인 설계 검증을 하기 위해서는 시뮬레이션 패턴을 적절히 선택해야 함을 나타내 준다. 그러나 일반적인 경우 시간에 대한 것을 고려하지 않아도 큰 회로에 대해 $||E||$를 구하는 것은 거의 불가능하다. 만일 시간을 고려하면 더욱 복잡해지게 된다. 더구나 위의 분석은 설계 오류가 조합회로를 순차회로로 또는 순차회로가 조합회로로 바뀌는 경우를 고려하지 않았다.

큰 회로를 다루기 위해 기본적인 설계 오류 모델이 사용된다. 그러나 설계 오류 모델은 고정적인 것이 아니고 설계자의 지식, 경험, 의도에 따라 추가될 수 있다. 설계 오류 모델이 정해지면 이를 이용해 검증율이 정해질 수 있다. 주어진 시뮬레이션 패턴의 효율성을 표현하는 시뮬레이션 검증율 (SCM)은 아래와 같이 표현된다.

SCM = 검출된 오류의 수 / 모델링

된 오류의 수 ×100[%]

이는 설계 오류의 수에 기초하므로 설계 검증의 척도를 더 정확하게 나타낸다.

3. 샘플링을 이용한 설계 오류 시뮬레이션

전형적인 설계 검증 방법은 시뮬레이션을 이용하는 것으로 사용자가 어느정도의 시뮬레이션 패턴을 준비하여 사용한다. 시뮬레이션이 끝난 후 시뮬레이션 결과에 만족하면 검증과정을 끝내게 되고 그렇지 않으면 더 많은 시뮬레이션 패턴을 가지고 만족한 결과를 얻을 때까지 계속 하게 된다. 이 과정에서 시뮬레이션의 계속여부를 결정하는 요소가 실제 얼마나 설계가 검증되었느냐가 아닐 수도 있다.

설계 오류모델링을 이용한 설계 검증 시스템의 장점은 시뮬레이션 검증율과 시뮬레이션 결과를 출력하는 것이다. 게다가 시뮬레이션 패턴을 생성하고 통계적 샘플링 방법을 이용해 검증율을 예측할 수도 있다. 이 시스템은 병렬 패턴 알고리듬[6]에 기초한 설계 오류 시뮬레이션과 자동 설계 오류 시뮬레이션 패턴 생성을 사용하고 있다. 이 시스템의 검증 방법은 아래와 같다. 설계와 주어진 모델의 명세에 따라 설계 오류 집합이 결정된다. 설계 오류 집합은 기본적인 설계 오류 모델(선 오류, 게이트 오류, 지역 오류, 포함 오류)[1]과 설계자가 제공하는 특정한 모델을 포함한다. 시뮬레이션 패턴은 사용자가 제공하거나 설계 오류 시뮬레이션 패턴 생성기로 부터 생성되어진다. 주어진 패턴에 대해 설계 오류 시뮬레이션을 수행하고 설계 오류 모델에 기초한 시뮬레이션 검증율을 출력한다. 사용자는 시뮬레이션 결과와 검증율에 따라 시뮬레이션의 지속 여부를 결정한다. 만일 만족할만한 검증율을 얻었으면 시뮬레이션을 멈추고 시뮬레이션 검증율이 충분하지 않다면 다른 패턴을 가지고 시뮬레이션을 계속한다. 샘플링을 사용하여 시뮬레이션 패턴의 일부와 설계 오류의 일부를 가지고 전체 시뮬레이션 검증율을 예측할 수 있다.

4. 샘플링 기법

설계 오류 시뮬레이션의 가장 큰 장점은 시뮬레이션 검증율을 제공하는 것이다. 이런 척도가 없다면 사용가능한 계산 자원이 고갈되거나 검증의 완성에 대한 정확한 지표가 아닌 감에 의해 시뮬레이션을 정지하게 된다. 설계 오류 시뮬레이션은 일반적으로 모델링된 설계 오류의 수에 따라 증가하므로 설계 오류 샘플링을 이용해 설계 오류 시뮬레이션의 비용을 줄이려 한다. 이를 위해 E를 모델링된 오류의 수라 하면 무작위로 샘플링된 E보다 작은 e만큼의 오류를 가지고 시뮬레이션을 수행하는 것이다. 따라서 설계 오류 시뮬레이션의 비용과 예측된 시뮬레이션 검증율 정확성과는 트레이드오프 관계에 있다. 문제는 어떻게 예측된 검증율이 실제 검증율에 근접하도록 샘플의 크기를 결정하는 것이다. 사용자가 설계에 많은 오류가 있음을 알거나 정확하지는 않지만 빨리 검증율을 얻고 싶을 때 사용할 수 있다.

샘플링 된 오류의 수 e는 무작위로 선택되기 때문에 검출된 오류의 수 d는 무작위 변수로 여겨질 수 있다. 원래 E개의 오류중에서 D개의 오류가 검출될 때 주어진 시뮬레이션 패턴이 e개의 샘플링된 오류중에서 d개의 오류를 검출할 확률은 아래와 같다.

$$\text{Prob} = \binom{D}{d}\binom{E-D}{e-d}\Big/\binom{E}{e}$$

이는 평균이 $\mu = e \times D/E$ 이고 분산이 $\sigma^2 = e \times D/E \times (1-D/E) \times (D-d)/(E-e) \approx eD(1-D)(1 - e/E)$인 hypergeometric 분포를 갖는다. 큰 회로에 대해 위의 분포는 위의 평균과 분산을 갖는 정규 분포를 갖는다. 따라서 예측된 시뮬레이션 검증율은 평균이 $\mu_e = \mu/e$이고 분산이 $\sigma_e^2 = \sigma^2/e^2$인 정규 분포를 갖는 무작위 변수이다. 이 경우에는 설계를 작은 단위로 분할해서 각 분할에서 오류를 선택하였다. 이렇게 하는 이유는 오류의 선택을 지역적으로 무작위하게 해서 고장의 경우보다 높은 정확성을 기하기 위함이다. 설계 오류의 일부가 고려된 후 사용자는 더 정확한 검증율을 얻기 위해 오류를 첨가하면 된다.

5. 결과

위의 시스템은 약 10,000줄의 C로 구성되어 있다. 설계 오류 시뮬레이션의 결과를 표 1에 나타내었다. 표 1에는 모델된 오류의수, 검출된 오류의 수, 시뮬레이션 검증율, 그리고 오류 시뮬레이션 시간이 나타나 있다. ISCAS 85 벤치마크 회로에 대해, 1024개의 무작위 패턴이 사용되었고 64Mb의 주메모리를 가진

SUN SPARC 20에서 수행하였다.

설계 오류 시뮬레이션의 샘플링 결과를 표 2에 나타내었다. 이 결과에는 표 1과 같은 1024개의 무작위 패턴이 사용되었다. 오류의 샘플링을 모델된 오류의 5%, 10%, 20%로 하였다. 예상했던대로 샘플의 크기가 클수록 더 정확한 예측과 시뮬레이션이 가능했고 더 많은 시뮬레이션 시간이 필요했다. 예를 들어 10%의 샘플링에 대해 평균 오차는 0.0204이고 20%의 샘플링에 대한 평균 오차는 0.0118이다. 결과는 주어진 시간과 메모리에서 예측이 가능함을 보여준다. 또한 시뮬레이션 패턴에 대한 무작위 샘플링뿐만 아니라 설계 오류에 대한 샘플링도 동시에 고려한다면 이 방법을 좀더 개선시킬 수 있을 것이다.

6. 결론

디지털 시스템의 크기가 커짐에 따라 이의 설계 및 검증은 더욱 어려워지고 있다. 위 시스템은 설계 오류 조건에 따라 회로를 시뮬레이션 할 수 있고 설계 검증 과정의 중요한 지표가 되는 시뮬레이션 검증율을 제공하고 샘플링 기법을 이용하여 시뮬레이션 검증율을 예측할 수 있게 한다. 따라서 이 시스템은 설계시간을 줄이는 효율적인 도구로 쓰일 수 있다.

참고문헌

[1] S. Kang and S. Szygenda, "Modeling and Simulation of Design Errors," in Proc. of ICCD, 1992, pp. 443-446.

[2] P. Chung and I. Hajj, "ACCORD: Automatic Catching and Correction of Logic Deisgn Errors in Combinational Circuits," in Proc. of ITC, 1992, pp. 742-751.

[3] S. Kang and S. Szygenda, "Automatic Error Pattern Generation for Deisgn Error Detection in a Design Validation," in Proc. of ASIC Conf., 1992, pp. 533-536.

[4] V. Agrawal et. al., "Test Generation by Fault Sampling," in Proc. of ICCAD, 1988, pp. 58-61.

[5] V. Agrawal, "Sampling Techniques for Determining Fault Coverage in LSI Circuits," in Journal of Digital Systems, 1981.

[6] J. Waicukauski et. al., "A Statistical Calculation of Fault Detection Probabilities by Fast Fault Simulation," in Proc. of ITC, 1985, pp. 779-784.

회로	모델링된 오류의 수	검출된 오류의 수	검증율 [%]	시간 [sec]
c2670	147824	120411	81.45	91.37
c3540	65578	63483	96.51	36.48
c5315	274502	266189	96.97	113.78
c6288	107376	105547	98.29	597.07
c7552	448099	425502	94.96	218.78

표1. 설계오류 시뮬레이션의 결과

회로	5%			10%			20%		
	검출된 오류수	검증율 [%]	시간 [sec]	검출된 오류수	검증율 [%]	시간 [sec]	검출된 오류수	검증율 [%]	시간 [sec]
c2670	6097	82.49	13.55	12166	82.30	17.75	24306	82.21	13.55
c3540	3211	97.66	4.22	6428	97.73	5.97	12814	97.41	9.55
c5310	13489	98.28	12.70	26939	98.14	17.87	53586	97.61	27.88
c6288	4355	80.76	31.47	9305	86.66	69.21	19782	92.12	129.60
c7552	21622	96.51	19.85	43196	96.40	29.83	85722	95.65	52.32

표2. 샘플링을 이용한 설계 오류 시뮬레이션의 결과