Simulation Cost Reduction Strategies for Behavioral Model Verification in Bayesian Based Stopping Rule

Kang Chul Kim[1], Chang-Gyoon Lim[1], Jae Hung Yoo[1], Seok Bung Han[2]

[1] Department of Computing Engineering, Chonnam National University,
San 96-1, Dunduck-Dong, Yeosu, Chonnam, Korea,
kkc@chonnam.ac.kr
[2] Department of Electronic Engineering, Gyeongsang National University,
Gajwa-Dong, Jinju, Gyeongnam, Korea
hsb@nongae.gsnu.ac.kr

Abstract. This paper presents two additional strategies to reduce simulation time for Bayesian based stopping rules in VHDL model verification. The first is that a semi-random variable is defined and the data staying in the semi-random variable range are skipped when stopping rule is running, and a turning point that can partition a random variable into a semi-random and a genuine random variable is chosen. The second is that the old values of parameters are kept when phases of stopping rule change. 12 VHDL models are examined, and the simulation results demonstrate that more than approximately 25% of clock cycles are reduced when using the two proposed strategies with 0.6% branch coverage rate loss.

Key Words : Verification, behavioral VHDL model, stopping rule, branch coverage, semi-random variable

1. Introduction

As VLSI fabrication technology has rapidly developed, it is possible to implement SoC(System on Chip) which comprises CPU, memory controller, bus controller, and so on. However, the technologies for the design and the verification of complicated chips have not developed rapidly, therefore, design verification of behavioral models is becoming difficult and has become a critical and time-consuming process in hardware design. Approximately 70% of the design effort in SoC, IP, ASIC, etc., has been consumed in the verification process, and design and verification engineers work together from the beginning of IC design [1,2,3].

It is impossible to know that the design being verified is indeed functionally correct with 100% certainty. That is the reason code coverage is required to be used. Many different types of coverage criteria have been defined including statement, branch, block, expression, and path coverage[4,5,6]. Among these criteria, branch coverage has been chosen as a good metric to verify hardware designs.

A large number of clock cycles should be consumed to verify a design by using coverage metrics because random test patterns have been used in complex VHDL models. Several examples demonstrate that 5 billion instruction simulation cycles can be run to ensure fault-free chip before tapeout[7,8,9]. Therefore, techniques that can seek the optimal stopping point should be developed and it is necessary to take productive stopping rules and strategies to reduce verification time and cost in a verification process.

This paper presents two additional strategies to reduce clock cycles for the Bayesian based stopping rule in VHDL model verification. In the first strategy, a semi-random variable is defined and the data located in the semi-random variable range are skipped when the stopping rule is running. A turning point partitioning the given random variable into a semi-random and a genuine random variable is chosen. In the second, the old values of parameters are kept when the phases of stopping rule change. More than approximately 25% of clock cycles are reduced using the two proposed strategies with few losses of branch coverage.

In section Ⅱ, previous work is introduced, and proposed strategies are explained in section Ⅲ. Section Ⅳ presents simulation results and discussions of the results. Finally, conclusion of this paper is presented in section Ⅴ.

2. Bayesian based stopping rules

Many methods for testing software program have been developed[10,11,12]. Recently, Poisson distribution and Bayesian estimation based stopping rules were introduced by [10] and [13].

[10] suggested a compounded counting process using empirical Bayesian principles and introduced the stopping rule in verification of VHDL programs. The key idea is to combine two probability distributions, that is, the number and the size of interruption. The parameters of probability distributions are assumed random variables by using Bayesian estimation.

[13] observed the outcome of branch coverage at every testing clock cycle and predicted the number of branches to be covered at time t. In this method, the number to be covered is expressed by branch coverage random process X_t. X_t is divided into two probabilities, interruption N_t, where one or more new branches are covered at time t, and the size of the interruptions W_t. The conditional distribution functions of the interruption occurrences, D_t, is defined. Next, a PMF(Probability Mass Function) extraction experiments are conducted in order to estimate the best fitted distribution function at every discrete time t, and Poisson probability distribution function is chosen. Then W_t is defined to be a random process distributed as a Poisson process with β_t as

$$W_t \sim e^{-\beta_t} \frac{\beta_t^{w-1}}{(w-1)!} . \qquad (1)$$

Here, β_t is a random variable representing the parameter of the Poisson distribution that should be estimated from the simulation history and is defined as $\beta_t = \beta g(t)$ for constant β and a decreasing function g(t) and $G(t) = \sum_{j:d_j=1} g(j)$.

The probability of having an interruption during the testing process, p(t), is estimated for every discrete time t. This p(t) is decomposed into a shape function f(t) and an amplitude value ς_t, p(t)= ς_t f(t), where the ς_t values can be determined statically or dynamically based on the history of testing the behavioral model. The gamma function of β is given by

$$\Gamma(\beta;\gamma,r) = \frac{\gamma^r}{\Gamma(r)} e^{-\gamma\beta} \beta^{r-1} . \qquad (2)$$

Then, the statistical model for the branch coverage increase for a given history of verification is derived. And Bayesian analysis is achieved by calculating the likelihood function of the Bayesian parameter, β_t, and given the verification history, the coverage is expected at time t>T as

$$\hat{\beta} = \frac{r + x_t - n_t}{\gamma + G(t)} , \qquad (3)$$

$$E\{W_t \mid \vec{x}\} = 1 + \frac{r + x_t - n_t}{\gamma + G(t)} g(t) . \qquad (4)$$

Finally, the total number of branches to be covered at future time t>T is predicted as

$$E\{X_t \mid \vec{x}\} = x_T + \sum_{j=T+1}^{t} (1 + \frac{r + x_t - n_t}{\gamma + G(t)} g(j)) \varsigma f(j) . \qquad (5)$$

In some papers[14,15], a new test pattern is generated at every simulation cycle and fed into a model, however [14] represents a new verification strategy. When data bits have to be fixed for certain times, a pattern for a certain number of clock cycles is used, that is, 1, 2, 4, 6 clock cycles are remained for each phase stage. The mixed strategy of random testing improves branch coverage[16].

The methods of above papers have improved behavioral model verification. If a few additional strategies are introduced, it is suggested that the verification cost can be reduced much further. A numerical result of an experiment can be a random variable when the value of it is not exactly expected. But the random variable used in [13] has a predictable value for some clock cycles from the beginning point. It is suggested that this random variable can be divided into two regions, that is, semi-random and genuine random variables. If the data in the semi-random variable range are skipped, the simulation time can be considerably reduced.

[16] did not use the previous simulation results when simulation phase changed. Bayesian estimation expects the future event by using the event estimated from the history of previous simulations, so it is reasonable to make use of the previous results in a current simulation.

In addition, [13] and [16] have the constraint that stops simulation when branch coverage is zero for first 30 consecutive clock cycles in every phase, but sometimes this can become a problem because the constraint can be of greater significance than stopping rules, and ultimately make the simulation stop.

3. Proposed strategies for reducing simulation cost

The overall expected value of X_t is the sum of all the expected sizes of interrupts after time T as showed in (5). X_t and W_t have been assumed to be random variables for all clock cycles. If x_T is large in (5), the expected value of X_t is large, that is, the larger the sum of all coverage before time T, the longer the stopping point. Therefore the sum of coverage before time T should be low in order to reduce the stopping point in the time axis.

Table 1. Cumulative branch coverage vs. clock cycle

	B C(Branch Coverage)											
S# t	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
1	35	35	58	47	47	69	58	47	35	58	49	93
2	53	92	111	94	94	153	111	94	92	111	92	166
3	54	95	123	102	102	156	123	102	95	123	127	218
4	58	111	123	133	133	178	123	133	111	123	129	228
5	68	140	126	145	145	210	126	145	140	126	132	231
6	68	143	139	146	146	214	139	146	143	139	136	238
7	68	148	142	146	146	216	142	146	148	142	139	239
8	68	151	170	146	146	222	170	146	151	170	139	239
9	68	151	172	154	154	222	172	154	151	172	144	244
10	68	152	182	159	159	222	182	159	152	182	148	249

The values of random variables are not known with certainty, that is, only a set of possible time and probability of the random variables are known. In this paper, a semi-random variable is defined as a variable of which value is certainly not known, but all values of the variable are greater than the reference value. Table 1 presents cumulative branch coverage for ten simulation clock cycles from the starting point for 12 sample VHDL models. Every branch coverage for a few clock cycles from t=1 is comparatively large.

Table 2. Branch coverage for 10 test patterns at the first two clock cycles.

	B C																							
S#	S1		S2		S3		S4		S5		S6		S7		S8		S9		S10		S11		S12	
t TP	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2	t=1	t=2
1	35	16	35	57	58	53	47	47	47	47	69	84	58	53	47	47	35	57	58	53	49	42	93	73
2	35	17	35	60	58	49	47	44	47	44	69	77	58	49	47	44	35	60	58	49	49	41	93	93
3	35	18	35	59	58	55	47	43	47	43	69	76	58	55	47	43	35	59	58	55	49	59	93	92
4	35	14	35	59	58	51	47	46	47	46	69	76	58	51	47	46	35	59	58	51	49	58	93	90
5	35	15	35	56	58	46	47	44	47	44	69	83	58	46	47	44	35	56	58	46	49	47	93	83
6	35	16	35	56	58	48	47	45	47	45	69	83	58	48	47	45	35	56	58	48	49	46	93	66
7	35	15	35	56	58	53	47	45	47	45	69	83	58	53	47	45	35	56	58	53	49	51	93	83
8	35	11	35	57	58	47	47	45	47	45	69	84	58	47	47	45	35	57	58	47	49	44	93	73
9	35	17	35	60	58	56	47	44	47	44	69	77	58	56	47	44	35	60	58	56	49	56	93	84
10	35	12	35	60	58	53	47	44	47	44	69	77	58	53	47	44	35	60	58	53	49	42	93	76

Table 2 presents that the number of branch coverage with 10 test patterns is comparatively high for first and second simulation clock cycles. Fig. 1 presents the histogram of branch coverage in table 2. The probability that the size of interruption is less than 10 is 0, and all the branch coverage is higher than 10 for the two consecutive clock cycles.

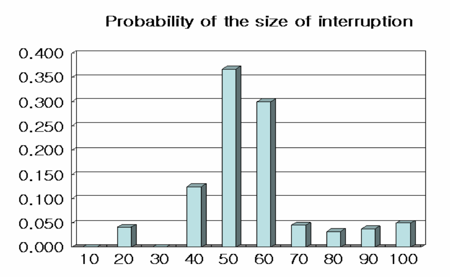

Fig. 1 Histogram of branch coverage for first and second clock cycles

Table 3 presents branch coverage at each cycle for 10 clock cycles with the same test pattern. For example, if the reference value has been chosen as 6, branch coverage is 35 and 18 at t=1 and t=2 respectively for sample S1, so the branch coverage in the time range between t=1 and t=2 defines a semi-random variable, and the branch coverage in the time range for t>2 defines a genuine random variable.

Table 3. Branch coverage for each clock cycle

	BC											
CC	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
1	35	35	58	47	47	69	58	47	35	58	49	93
2	18	57	53	47	47	84	53	47	57	53	43	73
3	1	3	12	8	8	3	12	8	3	12	35	52
4	4	16	0	31	31	22	0	31	16	0	2	10
5	0	29	3	12	12	32	3	12	29	3	3	3
6	0	3	13	1	1	4	13	1	3	13	4	7
7	0	5	3	0	0	2	3	0	5	3	3	1
8	0	3	28	0	0	6	28	0	3	28	0	0
9	0	0	2	8	8	0	2	8	0	2	5	5
10	0	1	10	5	5	0	10	5	1	10	4	5

The time is partitioned into two regions, TS and TR. TS is defined as TS={t_1, …,t_k }, where $W_t(t_i) \geq V_r$ for all i=1,2,..,k. and the next $W_t(t_{k+1}) < V_r$, where V_r is a reference value. Then the interval TR is defined as TR={t_{k+1}, …,∞}, where $W_t(t_i)$ can take non-negative value for all i=k+1,…, ∞.

It is important to note that values of the size of interruption W_t s are comparatively large for a few clock cycles from the beginning point even though the exact values of W_t s can not be predicted. W_t can be divided into two regions, that is, W_{ts} is a semi-random variable in the first region of a random variable and W_{tr} is a random variable in the other region as

$$W_t = W_{ts} \,||\, W_{tr} \,. \tag{6}$$

W_{ts} in the first part of a random variable is so large that the data in the semi-random variable range can be skipped when the stopping rule is running because W_t s are not predicted by the Poisson's distribution function. The semi-random variable is extremely important because the cumulative branch coverage in the semi-random variable nearly reaches half the total coverage though the first region has a few clock cycles. Therefore it is very important to make a reasonable choice for the turning point at a random variable because it has a great effect on the stopping point.

The dark dashed line presents the probability of W_t for S7 in Fig. 2[13]. If the value of W_t is greater than 6, the probability of N_t is quite low. It can be recognized that the probability that the values of W_t are greater than 6 is very small in the genuine random variable range, but the probability in the semi-random variable is always 1. The reference or turning point is decided to be the first value of W_t having the frequency more than 1% at first from the beginning point. In Fig.2, W_t of which branch coverage is less than 6 at first is chosen as a turning point, which divides a random variable into a semi-random and a genuine random variable. Depending on behavioral models and test patterns, the branch coverage is randomly varied as the number of clock cycle increases, therefore the position of the reference value can not be fixed in time region. Instead, this is dynamically decided while simulation is running.

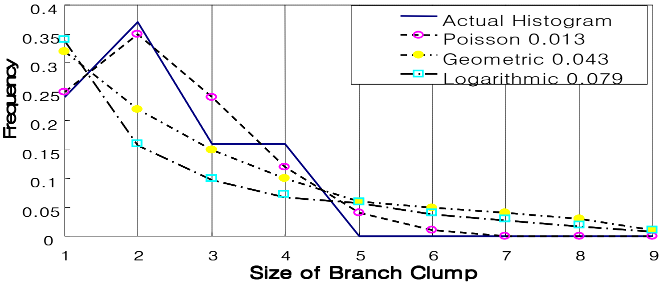

Fig. 2 Histogram fitting example of W_t for S7

The first additional strategy is proposed as follows. The branch coverage in the semi-random variable range W_{ts} defined in (6) is skipped while the stopping rule is running, then stopping point will become short because x_T becomes lower in (5). As the stopping point is shorter, the problem that total branch coverage becomes low must be considered. But random test patterns are used in most behavioral model verifications before testing a chip, and the increase in branch coverage rate abruptly drops as the number of random test pattern increases beyond the turning point, therefore it can be predicted that total branch coverage does not rapidly decrease when the data in the semi-random region W_{ts} is skipped.

The values of parameters obtained at the previous phases in (5) until time T are reset when a new phase starts in [13]. But the Bayesian model predicts the expected branch coverage using the previous branch coverage. The overall expected value of X_t

at any time t>T given the verification history up to time T is the sum of all the expected sizes of interruptions after time T. Thus the second additional strategy is proposed, where the old data obtained in the previous phase can be used as initial values in the next phase.

Applying the new verification strategies to the stopping rule of the Bayesian model can reduce clock cycles and improve performance of behavioral model verification.

4. Simulation results and discussions

In order to inspect the proposed algorithm, QuickVHDL simulator of Mentor Graphics is used. Fig. 3 explains the overall simulation process. The random test patterns are generated and the phase of test pattern is chosen, then simulation is conducted with the VHDL sample program and input files. The branch coverage is calculated during simulation, and the simulator decides whether the simulation continues or quitting by branch coverage and the proposed strategies.

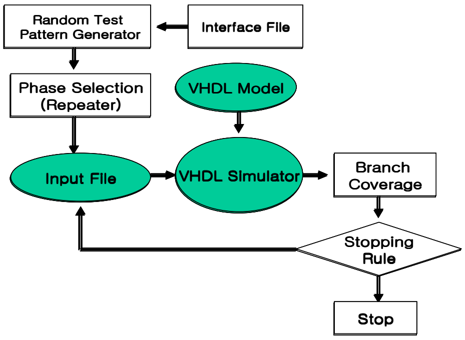

Fig. 3. Simulation process

12 VHDL models are examined to compare the effectiveness of proposed strategies with stopping rule of [14]. Table 4 presents the information of sample VHDL models.

Table 4. Sample VHDL programs used in simulation

Sample	Description	# of code line	#of branch
S1	16 megabit byte-wide top boot 150ns	1880	373
S2	CMOS syncBIFIFO 256x36x2	4657	251
S3	SyncFIFO with bus-matching 1024x36	5015	302
S4	SyncFIFO 2048x36	4667	225
S5	SyncFIFO 2048x36	4710	225
S6	CMOS syncBIFIFO 1024x36x2	4949	296
S7	SyncFIFO with bus-matching 1024x36	4963	302
S8	SyncFIFO 2048x36	4777	225
S9	CMOS syncBIFIFO 512x36x2	4752	251
S10	SyncFIFO with bus-matching 512x36	4973	302
S11	SyncBIFIFO with bus-matching 512x36x2	5498	399
S12	SyncBIFIFO with bus-matching 1024x36x2	5770	470

Table 5 presents 3 kinds of simulation results obtained while the proposed stopping rule is run with branch coverage of each clock cycle. In Table 5, SB is the result for static Bayesian estimator in [13]. SBF is the result for the proposed estimator, which is identical to SB except for two differences. The first is that the data of branch coverage that is greater than 6 from the beginning time are not used for the simulation because they stay in the semi-random variable range, and the second is that the result of previous phase in (5) has been used in the next phase simulation whenever phases change under the same condition of SB. Estimator SB30 is same to SBF except that the constraint that makes the simulation stop in a current phase is added to SBF whenever each branch coverage for 30 consecutive test patterns from an arbitrarily starting point is zero.

Compared to SB, clock cycles are reduced to 24.6% and 59.1% and branch coverage are reduced to 0.6% and 1.5% for SBF and SB30, respectively. Most branches are detected in the semi-random variable range and the coverage rate decreases as the number of clock cycles increases and the increase in coverage rate becomes very slow after the turning point. SB30 and SBF have 3 and 2 additional constraints compared to SB respectively as discussed in section Ⅲ, but obtain much better results than SB.

Table 5. Simulation results of SB, SB30 and SBF

Sample	SB[13]		SB30		SBF	
	CC	BC	CC	BC	CC	BC
S1	1854	128	530	128	1460	128
S2	631	199	493	204	1461	206
S3	808	232	495	231	485	231
S4	673	192	485	192	482	192
S5	789	195	485	195	482	195
S6	2249	273	490	247	489	249
S7	809	232	495	231	496	231
S8	2181	208	649	203	1456	208
S9	631	199	493	204	1461	206
S10	808	232	495	231	496	231
S11	1982	245	596	245	1460	247
S12	2080	364	628	348	1462	360
SUM	15,495	2,699	6,334	2,659	11,690	2,684

Estimator SB1 is the same as SBF, but the simulation does not stop even though the branch coverage is zero for 30 successive clock cycles from the starting point of phase Φ1, Φ2, Φ3, 4Φ. Simulation results of SB and SB1 for 4 phase stages are presented in table 6. Here, the branch coverage of Φ0 are results obtained before the turning point, that is, they are in a semi-random variable region, therefore the branch coverage is skipped while the stopping rule is operating. Let 30ZBC be zero branch coverage for 30 consecutive clock cycles from the beginning point. The numbers in CC column of SB are ones of clock cycles consumed for each phase, but the numbers in CC column of SB1 means cumulative clock cycles for each phase including Φ0. The total clock cycles of SB are much less than one of SB1 in S9 because the simulation of S9 under SB stops by 30ZBC constraints in phase Φ2, Φ4, but SB suffers from a drawback that the cumulative branch coverage is relatively low. The cumulative clock cycles of SB in S1 and S12 is much larger because they are not affected by 30ZBC constraint. SB1 in S1, S9, and S12 has no 30ZBC constraint, but the cumulative clock cycles are 1462 on average with good branch coverage. It can be concluded that the stopping points are decided by the proposed strategies.

Table 6. Comparison of SB and SB 1 for S1,S9, and S12

PH	S1				S9				S12			
	SB		SB1		SB		SB1		SB		SB1	
	CC	BC	CC	BC	CC	BC	CC	BC	CC	BC	CC	BC
Φ0			2	53			2	92			4	228
Φ1	94	121	88	121	99	183	88	177	112	336	90	336
Φ2	188	1	161	122	60	0	161	182	60	0	163	336
Φ3	120	0	306	122	292	16	308	192	324	10	308	346
Φ4	1452	6	1460	128	180	0	1461	206	1584	18	1462	360
TOT	1854	128	1460	128	631	199	1461	206	2080	364	1462	360

It is difficult to obtain 100% branch coverage because random test patterns are applied to the simulator. The effect of semi-random variable is hidden because some simulation results have 30ZBC so that the simulation stops by 30ZBC constraints. Simulation results of Table 5 that do not have 30ZBC in phase Φ4 are selected and represented in Table 7 again to consider the effect of the proposed strategies. The simulation results of SBF reduce 2259 clock cycles with losses of 2 branch coverage because SBF skips the branch coverage detected in the W_{ts} region.

Table 7. Comparisons of SB and SBF without constraint of 30ZBC

Sample	SB		SBF	
	CC	BC	CC	BC
S1	1854	128	1460	128
S8	2181	208	1456	208
S11	1982	245	1460	247
S12	2080	364	1462	360
SUM	8097	945	5838	943

5. Conclusion

This paper proposes strategies that can be used to reduce clock cycles in behavioral model verification. The techniques are applied to 12 sample programs and the simulation results demonstrate that a large number of clock cycles are reduced with a few losses in branch coverage. In the future, finding the trade-off of branch coverage vs. consumption of clock cycle and an efficient turning point that can further reduce the number of simulation clock will be studied.

References

1. Frank Vahid, Tony Givargis. : EMBEDDED SYSTEM DESIGN : A Unified

Hardware/Software Introduction. John Wiley & Sons, Inc. (2002)

2. Samiha Mourad, Yervant Zorian. : Principles of Testing Electronic Systems. John Wiley & Sons, Inc. (2000)

3. Janick Bergerson.: Writing testbenches : functional verification of HDL model. Kluwer Academic (2003)

4. Jjm Lipman.: Covering your HDL chip-design bets. EDN, (Oct. 22, 1998) 65-70

5. Brian Barrera, “Code coverage analysis-essential to a safe design,” Electronic Engineering, (Nov. 1998) 41-43

6. Kevin Skahill, “A designer’s guide to VHDL design and verification,” Engineering Software, (Feb. 1996) 149-158

7. A. Von Mayrhauser, et. Al., “On choosing test criteria for behavioral level hardware design verification”, IEEE International Hign Level Design Validation and Test Workshop, Berkeley, CA, (2000)

8. A. Hajjar, Chen T, Improving the efficiency and quality of simulation-based model verification using dynamic Bayesian criteria, Quality Electronic Design, Proceedings International Symposium, (March 2002) 304-309

9. J. Gately, “Verifying a million gate processor”, Integrated System Design, (1997) 19-24

10. M. Sahinoglu, A. Von Mayrhauser, A. Hajjar, T. Chen, C. Anderson, “ On the efficiency of a compound Poisson stopping rule for mixed strategy testing”, IEEE Aerospace conference, Track 7, (Mar. 1997)

11. W. Howden, “ Confidence-based reliability and statistical coverage estimation”, Proceedings on the international symposium on software reliability engineering, (Nov. 1997) 283-291.

12. S. Chen, S. Mills, “A binary markov process model for random testing”, IEEE transactions on software engineering, v22, n3, (Mar. 1996) 218-223

13. Amjad Fuad A. Hajjar, “Bayesian Based Stoping Rules for Behavioral VHDL Verification”, Ph. D Dissertation, Fall 2000.

14. R. Ramchandani, D. Thomas, “Behavioral test generation using mixed integer nonlinear programming”, International test conference, (1994) 958-967

15. G. Hayek, C. Robach, “From specification validation to hardware testing: a unified method”, International Test conference, (1996) 885-893

16. T. Chen, M. Sahinoglu, A. Mayrhauser, A. Hajjar, A. Anderson, “Achieving the quality for behavioral models with minimum effort”, 1st international symposium on quality in electronic design, (Mar. 2000)