Using Range-equivalent Circuits for Facilitating Bounded Sequential Equivalence Checking

Yung-Chih Chen[1], Wei-An Ji[2], Chih-Chung Wang[2], Ching-Yi Huang[2], Chun-Yao Wang[2]
[1]Department of Computer Science and Engineering, Yuan Ze University, Chungli, Taiwan, R.O.C.
[2]Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.

Abstract—This paper presents a method based on range-equivalent circuit technique for SAT-based bounded sequential equivalence checking. Given two sequential circuits to be verified, instead of straightforward unrolling the miter of two sequential circuits, we iteratively minimize the miter with a range-equivalent circuit technique before adding a new timeframe. This is because the previous timeframes can be considered as a pattern generator that feeds input patterns to the next timeframe. Experimental results show that the proposed method saved up to 91% of time for reaching the same bounded depth compared with previous work on IWLS2005 benchmarks.

I. Introduction

Bounded sequential equivalence checking (BSEC), which is a special case of bounded model checking [2] [3] [4], verifies the functional equivalence of two sequential circuits with a bounded depth k. Traditionally, given two sequential circuits to be verified, the two circuits are first unfolded to k timeframes. Then, these two unfolded circuits are straightforward verified by using an equivalence checker. In general, the verification complexity strongly depends on the value k.

Since the unfolded circuits are combinational circuits, most combinational equivalence checking could be applied to improve the efficiency of BSEC [13] [17] [19]. Additionally, the work in [14] proposed a framework to explore the structural similarities between two circuits under verification and merge the similarities to reduce the verification complexity. In [6] [20], the verification problem of two unfolded circuits was formulated as a satisfiability (SAT) problem and some learned constraints on logic dependencies were used to speedup the SAT solving process. Additionally, in [11], the authors used logic optimization methods [8] [9] [10] to minimize and restructure the unfolded circuits while preserving their equivalence or non-equivalence to reduce the verification complexity and facilitate the SAT solving process.

This work is supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 103-2221-E-007-125-MY3, MOST 103-2221-E-155-069, NSC 102-2221-E-007-140-MY3, and NSC 100-2628-E-007-031-MY3.

In general, when we verify the equivalence of two sequential circuits with a bounded depth k, we usually have confirmed that these two sequential circuits are equivalent from the 1^{st} to the $(k-1)^{th}$ timeframe. This known condition actually can be used to reduce the complexity of equivalence checking process at the k^{th} timeframe.

Based on this observation, we propose a method for BSEC model optimization as follows. The unfolded subcircuit of prior timeframes is considered as a pattern generator that feeds input patterns to the subcircuit of the k^{th} timeframe. Therefore, we only care about the outputs from this pattern generator. As a result, we optimize this pattern generator by using the function-preserving logic optimization techniques [11] [15] and the range-equivalent circuit technique [7] [16] before conducting the equivalence checking. By replacing the subcircuit of the 1^{st} to the $(k-1)^{th}$ timeframe with a smaller range-equivalent circuit, the equivalence checking process would be more efficient or reach a larger depth k.

The experiments were conducted on a set of IWLS 2005 benchmarks [21]. The experimental results show that the proposed approach can save up to 91% of verification time for reaching the same bounded depth k among all the benchmarks compared with the state-of-the-art [11].

The contributions of this work are two-fold: First, instead of unrolling k timeframes in the BSEC model at a time, this work proposes a verification flow that checks the equivalence iteratively in a divide-and-conquer manner. Second, this is the first work that exploits the range-equivalent circuit optimization technique to simplify the BSEC model for accelerating the verification process.

II. Preliminaries

A. SAT-based BSEC

The basic idea of the SAT-based BSEC is to formulate the equivalence checking of two unfolded circuits with k timeframes as a Boolean SAT problem and solve it with a SAT solver. Let us use an example in Fig. 1 to illustrate the process of SAT-based BSEC. Given two sequential circuits, P and Q, they are first constructed as a miter [5] by connecting their corresponding POs with additional XOR gates and connecting these XOR gates to an OR gate as shown in Fig. 1(a). Then, the miter is unfolded to

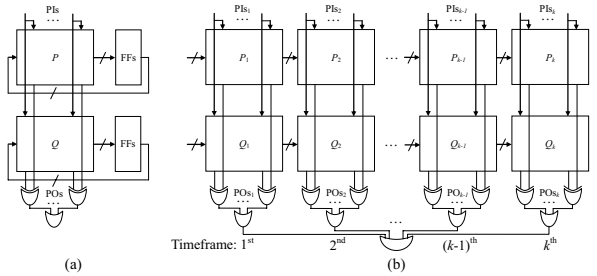

Fig. 1. An example of BSEC [11]. (a) The miter. (b) A BSEC model.

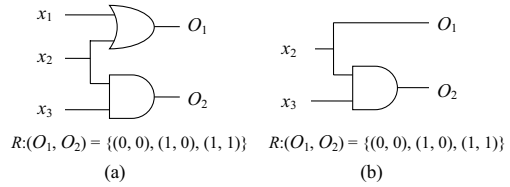

Fig. 2. An example of range-equivalent circuit optimization [7]. (a) The original circuit. (b) The resultant circuit by replacing x_1 with 0.

k timeframes where k is the bounded depth, and all the inserted OR gates of each timeframe are connected to an additional big OR gate as shown in Fig. 1(b). Finally, the output value of the big OR gate determines the functional equivalence of P and Q within the bounded depth k. If this output value can be evaluated as 1, P and Q are non-equivalent. This is because there exists at least one input pattern that generates different output values on at least one pair of the POs of P and Q within k timeframes. Conversely, if the output value is 0 for all input patterns, P and Q are functionally equivalent within k timeframes. To determine whether or not P and Q are equivalent within k timeframes, the BSEC model can be transformed into a conjunctive normal form (CNF) formula [18], and then be solved by using a SAT solver [12].

B. Range-equivalent Circuit

The *range* of a combinational circuit is the set of all output combinations that it can generate. For example, the range of the circuit in Fig. 2(a) is $(O_1, O_2) = \{(0,0),(1,0),(1,1)\}$. Range-equivalent circuit optimization is a technique that simplifies a circuit while preserving its range. For example, in Fig. 2(b), the range of the circuit is $(O_1, O_2) = \{(0,0),(1,0),(1,1)\}$ as well. Thus, Fig. 2(a) can be seen as the given circuit and Fig. 2(b) is the resultant circuit simplified by a range-equivalent circuit optimization technique [7].

III. An Overview

In this section, we explain the intent of the proposed method for enhancing SAT-based BSEC. In Fig. 1(b), if that P and Q are functionally equivalent within $(k-1)$ timeframes is a known condition, we do not need to connect the pairs of POs in the timeframes of 1 to $(k-1)$ to the additional XOR gates. That is, we can focus on verifying the POs in the k^{th} timeframe of P and Q, i.e., P_k and Q_k. In this way, we can minimize the circuit by removing the dangling gates after removing the POs. Fig. 3(a) shows the simplified BSEC model, in which only the POs of P_k and Q_k are connected to XOR gates as suggested.

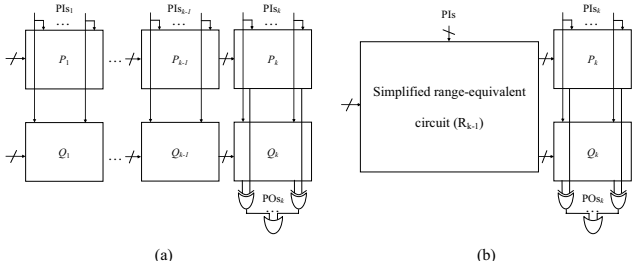

Fig. 3. Simplified BSEC models.

Additionally, to construct a more simplified BSEC model, we can apply a range-equivalent circuit optimization technique on pseudo primary outputs (PPOs) of each timeframe to minimize the timeframes of 1 to $(k-1)$, and then add the k^{th} timeframe to the minimized circuit. Here, please note that the PPOs are the signals which drive the next timeframe. We can further simplify the BSEC model while preserving the range of subcircuit consisting of R_{k-1} without affecting the verification result, where R_{k-1} is the range of unfolded timeframes of 1 to $(k-1)$, as shown in Fig. 3(b). Since a range-equivalent circuit usually has fewer PIs and gates, the size of the BSEC model can be further reduced. As a result, the verification process can be facilitated.

The key idea of this work is to use the known condition that two circuits are functionally equivalent within the first $(k-1)$ timeframes to enhance the equivalence checking of the k^{th} timeframe by using the range-equivalent circuit optimization technique [7].

IV. Enhanced SAT-based BSEC

In this section, we present the proposed algorithm for enhancing the SAT-based BSEC and use a simple example to demonstrate our ideas. In this example, assume that we would like to check the equivalence of two sequential circuits, P and Q, as shown in Fig. 4(a) with a bounded depth of 2. We first construct a miter by connecting their POs, O and O', with an XOR gate and simplify the circuit by using logic optimization methods. Now we have obtained a smaller circuit in one timeframe as shown in Fig. 4(b), which is used in the construction of BSEC model.

In this example, we assume that the initial state is $(I_1, I_2, I_3) = (0,0,0)$ for simplicity. Then, we check the output of the XOR gate with a SAT solver in Fig. 4(b). Once the output value of the XOR gate cannot be evaluated as 1 after running the SAT solver, P and Q are func-

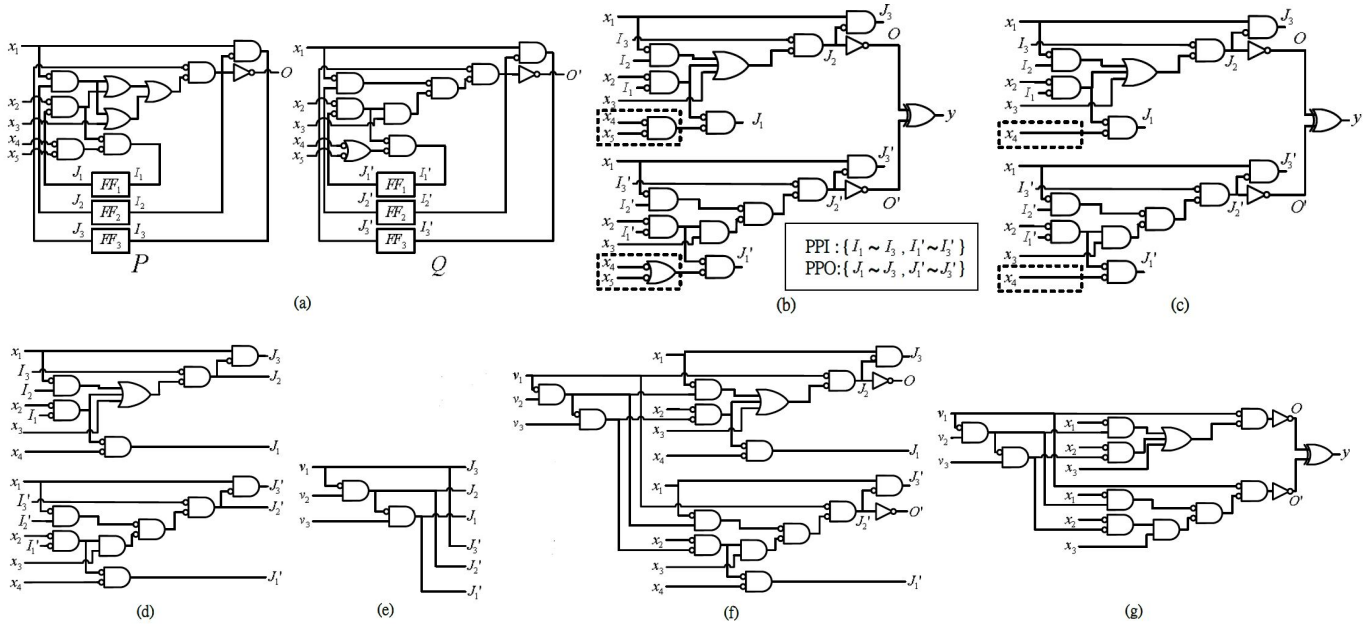

Fig. 4. An example for the proposed algorithm.

tionally equivalent for the 1^{st} timeframe. Next, since the 1^{st} timeframe subcircuit can be considered as a pattern generator to the 2^{nd} timeframe, we minimize this subcircuit by using the range-equivalent circuit optimization technique with respect to all the PPOs. The resultant circuit is as shown in Fig. 4(c), where the differences between Figs. 4(b) and 4(c) are highlighted. We can see that the subcircuits related to x_4 and x_5 are simplified as a single PI x_4.

Next, we remove the XOR gate (PO) of the BSEC model since we currently do not need this subcircuit. Some dangling gates related to the PO can be removed as well. Thus, the resultant circuit is as shown in Fig. 4(d) where two inverter gates are removed. With the assumption of the initial state, we obtain a more simplified range-equivalent circuit as shown in Fig. 4(e), where $v_1 \sim v_3$ are PIs, J_i and J_i' are outputs for driving the 2^{nd} timeframe.

To verify the equivalence of these two circuits for the 2^{nd} timeframe, we exploit the known fact that the 1^{st} timeframe has been equivalent. The resultant BSEC model after connecting to the 2^{nd} timeframe is shown in Fig. 4(f). Again, we only leave the PO subcircuit and connect O and O' with an XOR gate for verification. The resultant BSEC model is shown in Fig. 4(g). Since the output value of the XOR gate cannot be evaluated as 1 as well after running the SAT solving process, P and Q circuits are functionally equivalent with a bounded depth 2.

In summary, the overall flow of the proposed algorithm for BSEC model optimization is shown in Fig. 5. The inputs are two sequential circuits to be verified and a bounded depth k. The construction of optimized BSEC

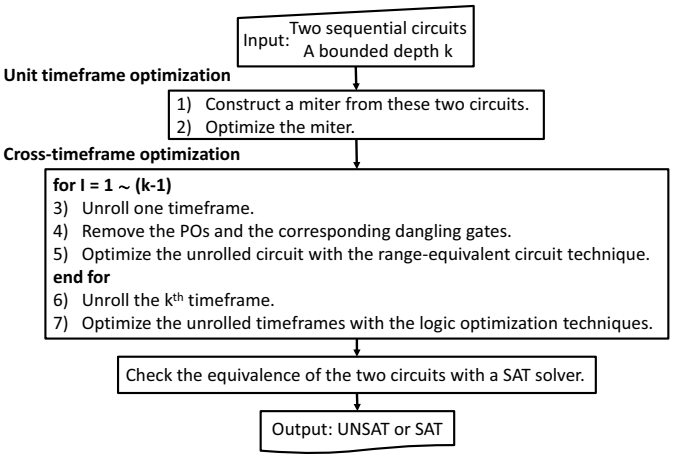

Fig. 5. The overall flow of BSEC model optimization.

model consists of two parts, and they are unit timeframe optimization, and cross-timeframe optimization. In the unit timeframe optimization, we connect the two circuits under verification as a miter, and then minimize the miter with the logic optimization techniques. In the cross-timeframe optimization, we unroll the minimized circuit to construct the BSEC model from the 1^{st} timeframe to the $(k-1)^{th}$ timeframe iteratively. In every timeframe of unrolling, we remove the POs and the corresponding dangling gates, and optimize the unrolled circuit with the range-equivalent circuit technique. Finally, we unroll the k^{th} timeframe and use logic optimization techniques again for all the unrolled timeframes to obtain a more minimized BSEC model. At the end, we check the equivalence of the

TABLE I
EXPERIMENTAL RESULTS OF SAT-BASED BSEC WITH A FIXED BOUNDED DEPTH.

Benchmark	FFs	k	[11]				Our Approach					Saved (%)
			PI	Gate	T_SAT	T_Total	PI_Red	T_Range	Gate	T_SAT	T_Total	
b04	**132**	**100**	**1100**	**41100**	**25.20**	**193.37**	**204**	**0.65**	**12270**	**0.01**	**15.97**	**91.74**
ss_pcm	174	45	855	18641	53.21	61.28	95	0.29	16661	45.52	60.48	1.30
usb_phy	196	40	600	15636	89.39	155.42	126	0.30	13358	16.14	75.45	51.45
sasc	234	40	640	9516	21.89	51.62	40	0.54	9403	4.90	38.09	26.21
s5378	328	26	910	12766	153.86	201.24	125	0.27	4406	3.00	16.16	91.96
s9234	422	20	720	7314	52.56	78.87	131	0.2	5782	31.57	51.46	34.75
spi	458	7	419	17084	807.05	937.24	20	0.05	15838	129.99	250.33	73.20
b14	**490**	**7**	**224**	**9545**	**68.50**	**131.78**	**0**	**0.02**	**8542**	**32.47**	**84.99**	**35.50**
b20	980	7	224	12442	584.22	664.75	32	0.07	11342	396.60	472.73	28.88
b22	**1470**	**5**	**160**	**6836**	**238.56**	**310.11**	**0**	**0.09**	**5218**	**25.99**	**96.91**	**68.74**
i2c	256	20	380	17511	954.66	1052.82	42	0.15	15603	589.91	678.87	35.51
s1494	673	50	400	52163	118.23	143.11	54	0.49	26010	53.15	71.97	49.70
b08	155	100	900	24704	183.25	194.22	8	1.13	21513	139.62	164.80	15.14
aes_core	**1060**	**6**	–	–	–	**MEM**	**15**	**10.24**	**93052**	**8612.53**	**9576.22**	–
s13207	**1338**	**350**	–	–	–	**MEM**	**8725**	**42.73**	**36929**	**32.49**	**1355.83**	–
Average			–	–	–	–	–	–	–	–	–	46.46

two circuits with a SAT solver.

V. EXPERIMENTAL RESULTS

We implemented the proposed method in C language within ABC [1]. For comparison, we also re-implemented the previous work [11], which also facilitated BSEC by minimizing the BSEC model. Note that the experiments in this work intend to demonstrate if the integration of range-equivalent circuit technique in the BSEC model construction speedups the verification process. Although we adopt the algorithm in [7] in our implementation, the algorithms in [16] are applicable as well.

The experiments were conducted on a 3.0 GHz Linux platform (CentOS 4.8) with 32 GBytes memory for a set of IWLS 2005 benchmarks [21]. The revised circuit for equivalence checking in the experiments was obtained by using the *resyn2* script [15], which is a logic optimization script in ABC. Other optimization methods can be also used to obtain the revised circuits. MiniSat [12] was used as the SAT solver. We conducted two experiments to demonstrate the effectiveness and efficiency of the proposed approach. In the first experiment, we show the CPU time of [11] and ours for reaching the same bounded depth k. We set different values of k for different benchmarks to show the benefits of the proposed method.

The experimental results are summarized in Table I. For example, *b04* has 132 FFs. In [11], the resultant model has 41100 gates and the total CPU time is 193.37s, where 25.20s are spent on SAT solving. As for our approach, we spend 0.65s to reduce 204 PIs from 1100 PIs with the range-equivalent circuit technique such that the gate count is reduced to 12270 after unrolling 100 times. As a result, our approach totally spends 15.97s, or 91.74% CPU time is saved compared with [11].

Note that although the experimental results of benchmarks *b14* and *b22* have no reduction on the number of PIs, their gate counts are still reduced as well. This is because we remove the unnecessary POs and the dangling gates after knowing the equivalence of prior timeframes. As a result, their CPU time savings are 35% and 68%, respectively.

For the circuits with thousands of FFs, like *aes_core* and *s13207*, [11] cannot successfully deal with them under the pre-determined k values due to memory explosion. However, our approach is still capable of verifying the circuits.

According to Table I, we find that the proposed approach is more efficient than [11]. The time saving ratios are in the range from 1% $\sim$ 91% among all the benchmarks, and the average time saving ratio is 46%.

In the second experiment, we show the maximal bounded depth k that each method reached within a CPU time limit, 36000s in Table II. For example, *sasc* has 234 FFs. [11] spends 20857s for verifying the equivalence at $k = 288$, while our approach can check the equivalence at $k = 325$ in 29523s. However, when $k = 289$, [11] exceeds the time limit.

For *b04* and *s5378*, the results show that our approach can reach much larger bounded depth k. This is because the number of PIs and the gate counts of these circuits can be reduced a lot. For *ss_pcm*, *b20*, and *b22*, the results are constrained by the memory usage rather than CPU time. Therefore, the programs are terminated due to memory explosion without exceeding the time limit. On the other hand, for some larger circuits, like *aes_core*, if k is increased by 1, the CPU time could exceed the

TABLE II
EXPERIMENTAL RESULTS WITH A 36000S TIME LIMIT.

Benchmark	FFs	[11]		Our Approach		Increase of k
		k	Time	k	Time	
b04	**132**	**540**	**18451**	**3565**	**32345**	**3025**
ss_pcm	**174**	**304**	**4359**	**513**	**28962**	**209**
usb_phy	196	232	26553	769	24561	537
sasc	**234**	**288**	**20857**	**325**	**29523**	**37**
s5378	**328**	**73**	**22951**	**2589**	**27569**	**2516**
s9234	422	125	16292	223	24589	98
spi	458	16	22563	33	19625	17
b14	490	82	12562	132	17598	50
b20	**980**	**15**	**10252**	**15**	**9854**	**0**
b22	**1470**	**5**	**310**	**26**	**12658**	**21**
i2c	256	125	23546	168	25689	43
s1494	673	301	19652	369	20356	68
b08	155	278	15950	342	23222	64
aes_core	**1060**	**5**	**575**	**6**	**9576**	**1**
s13207	1338	344	4866	432	13208	88

time limit for both approaches, or even terminated due to memory explosion. According to Table II, we find that our approach achieves a larger bounded depth for most benchmarks compared to [11] within a CPU time limit.

VI. CONCLUSION

In this paper, we propose an approach for facilitating SAT-based BSEC. The main idea is using the known condition that two circuits under verification are equivalent within $(k-1)$ timeframes to facilitate their equivalence checking at the k^{th} timeframe. The experimental results show that the proposed method can save up to 91% CPU time compared with the state-of-the-art and is capable of verifying large sequential circuits.

REFERENCES

[1] Berkeley Logic Synthesis and Verification Group, "ABC: A System for Sequential Synthesis and Verification," http://www.eecs.berkeley.edu/ alanmi/abc/.

[2] A. Biere, et al.,"Symbolic Model Checking Using SAT Procedures instead of BDDs," in *Proc. DAC*, 1999, pp. 317-320.

[3] A. Biere, et al., "Bounded Model Checking," *Advances in Computers*, vol. 58, 2003, pp. 117-148.

[4] A. Biere, et al., "Symbolic Model Checking without BDDs," in *Proc. Tools and Algorithms for the Construction and Analysis of Systems*, 1999, pp. 193-207.

[5] D. Brand, "Verification of Large Synthesized Designs," in *Proc. ICCAD*, 1993, pp. 534-537.

[6] Lynn C. L. Chang, et al., "Speeding up Bounded Sequential Equivalence Checking with Cross-Timeframe State-Pair Constraints from Data Learning," in *Proc. ITC*, 2009, pp. 1-8.

[7] Y. C. Chen, et al., "An Implicit Approach to Minimizing Range-Equivalent Circuits," *IEEE TCAD*, vol. 27, pp. 1942-1955, Nov. 2008.

[8] Y. C. Chen, et al., "Fast Detection of Node Mergers using Logic Implications," in *Proc. ICCAD*, pp. 785-788, 2009.

[9] Y. C. Chen, et al., "Fast Node Merging with Don't Cares Using Logic Implications," *IEEE TCAD*, vol. 29, pp. 1827-1832, Nov. 2010.

[10] Y. C. Chen, et al., "Node Addition and Removal in the Presence of Dont Cares," in *Proc. DAC*, pp. 505-510, 2010.

[11] Y. C. Chen, et al., "Logic Restructuring Using Node Addition and Removal," *IEEE TCAD*, vol. 31, pp. 260-270, Feb. 2012.

[12] Niklas Een, et al., "MiniSat - A SAT Solver with Conflict-Clause Minimization," in *Proc. SAT*, 2005.

[13] E. I. Goldberg, et al., "Using SAT for Combinational Equivalence Checking," in *Proc. DATE*, 2001, pp. 114-121.

[14] S. Y. Huang, et al., "AQUILA: An Equivalence Checking System for Large Sequential Designs," *IEEE TC*, vol. 49, pp. 443-464, May 2000.

[15] A. Mishchenko, et al., "DAG-Aware AIG Rewriting: A Fresh Look at Combinational Logic Synthesis," in *Proc. DAC*, 2006, pp. 532-536.

[16] J. Moondanos, et al., "CLEVER: Divide and Conquer Combinational Logic Equivalence VERification with False Negative Elimination," in *Proc. CAV*, 2001, pp. 131-143.

[17] S. Reda, et al., "Combinational Equivalence Checking Using Boolean Satisfiability and Binary Decision Diagrams," in *Proc. DATE*, 2001, pp. 122-126.

[18] G. S. Tseitin, "On the Complexity of Derivation in Propositional Calculus," in *Studies in Constr. Math. and Math. Logic*, 1968. pp. 115-125.

[19] A. Veneris, et al., "Logic Verification based on Diagnosis Technique," in *Proc. ASPDAC*, 2003, pp. 538-543.

[20] W. Wu, et al., "Mining Global Constraints with Domain Knowledge for Improving Bounded Sequential Equivalence Checking," *IEEE TCAD*, vol. 27, pp. 197-201, Jan. 2006.

[21] http://iwls.org/iwls2005/benchmarks.html