Formal Verification Coverage: Are the RTL-Properties Covering the Design's Architectural Intent?

Prasenjit Basu [*] Sayantan Das[*] Pallab Dasgupta[*] P.P. Chakrabarti[*]
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur, India.

Chunduri Rama Mohan[*]
DA Strategic Planning,
Intel Corporation, Folsom, USA.
chunduri.r.mohan@intel.com

Limor Fix[*]
Logic and Validation Technology
Intel Corporation, Haifa, Israel.
limor.fix@intel.com

1. Introduction

It is essential to formally ascertain whether the RTL validation effort effectively guarantees the correctness with respect to the design's architectural intent. The design's architectural intent can be expressed in formal properties. However, due to the capacity limitation of formal verification, these architectural-properties cannot be directly verified on the RTL. As a result, a set of lower level RTL-properties are developed and verified against the RTL. In this paper we present: (1) a method for checking whether the RTL-properties are covering the architectural-properties, that is, whether verifying the RTL-properties guarantee the correctness of the design's architectural intent, and (2) a method to identify the coverage holes in terms of the architectural-properties (or their sub-properties) that are not covered.

2. The notion of coverage

In our framework of reasoning, we assume that the architectural intent of a design, D, is given as a set of *high level* temporal properties over a set of architectural level signals, $\mathcal{AP}_A(D)$. We shall denote the conjunction of these properties by $\mathcal{A}^D$ (for a design D), and call them the *architecture spec* or *ASpec*. We are also given a set of temporal RTL *module properties* over the signals, $\mathcal{AP}_I(D)$, of the component modules of the design D. These properties will collectively be called the *implementation spec* or *ISpec* and will be denoted by $\mathcal{I}^D$. In this paper we address the problem of determining whether the *ISpec* covers the *ASpec*, and if not, finding the coverage gaps.

Example 1 Let us consider the design of an arbiter that arbitrates between two request lines r_1 and r_2. Let the corresponding grant lines be g_1 and g_2. Let one of the high level design requirements be that *the waiting time for any request to be served should not exceed 2 cycles*. We can express this requirement as the following two properties in Linear Temporal Logic (LTL) [1]:

$$\begin{array}{ll} F_1: & G(\,(r_1 \,\wedge\, X r_1) \,\Rightarrow\, X g_1 \,\vee\, XX g_1\,) \\ F_2: & G(\,(r_2 \,\wedge\, X r_2) \,\Rightarrow\, X g_2 \,\vee\, XX g_2\,) \end{array}$$

The first property states that *whenever r_1 is asserted in two successive cycles (say t_i and t_{i+1}), g_1 must be asserted in t_{i+1} or t_{i+2}*. The second property expresses a similar intent for r_2. The properties F_1 and F_2 belong to the ASpec.

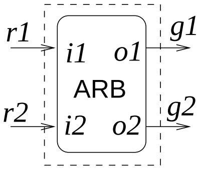

Figure 1. Sample design

Let us now consider an implementation of the arbiter with an existing arbiter module ARB as shown in Fig 1. The module ARB is known to satisfy the following properties:

$$\begin{array}{ll} I_1: & G(\,i_1 \,\Leftrightarrow\, X o_1\,) \\ I_2: & G(\,(i_2 \wedge \neg i_1) \,\Leftrightarrow\, X o_2\,) \end{array}$$

In other words, the module ARB gives i_1 higher priority than i_2. When i_2 is the sole request, then o_2 is asserted (in the next cycle). The properties I_1 and I_2 belong to the ISpec. We also have the properties that map the ASpec to the ISpec, namely:

$$\begin{array}{ll} I_3: & G(\,(r_1 \,\Leftrightarrow\, i_1) \wedge (r_2 \,\Leftrightarrow\, i_2) \\ & \quad \wedge (g_1 \,\Leftrightarrow\, o_1) \wedge (g_2 \,\Leftrightarrow\, o_2)\,) \end{array}$$

[*] The authors acknowledge Intel Corp. for partial support of this work.

1530-1591/04 $20.00 (c) 2004 IEEE

Our primary coverage problem is to determine whether $(I_1 \wedge I_2 \wedge I_3) \Rightarrow (F_1 \wedge F_2)$ is a tautology. In this case, the answer is negative. Consider the counter-example scenario where both r_1 and r_2 are asserted for 2 consecutive cycles (say t_i and t_{i+1}). The module ARB will assert only o_1 (and hence, g_1) in both t_{i+1} and t_{i+2}, which will refute F_2.

If the answer to our primary coverage problem is negative, then our secondary task is to identify the *coverage holes*. There are two main challenges in this task. The first is to find the coverage holes. The second is to present the coverage holes to the designer in a meaningful and legible way.

In this example, we can show by deduction that $(I_1 \wedge I_3) \Rightarrow F_1$. Therefore the uncovered ASpec lies in F_2. At this point we can present F_2 to the designer and indicate that the implementation fails to cover this property. However, we can do better. For example, consider the following decomposition of F_2:

F_{2a}: $G(\ ((r_2 \wedge Xr_2) \wedge (r_1 \wedge Xr_1)) \Rightarrow Xg_2 \vee XXg_2\ )$
F_{2b}: $G(\ ((r_2 \wedge Xr_2) \wedge \neg(r_1 \wedge Xr_1)) \Rightarrow Xg_2 \vee XXg_2\ )$

It can be shown (by deduction) that F_{2b} is covered by $(I_2 \wedge I_3)$. Therefore the uncovered ASpec is in F_{2a}. Presenting F_{2a} to the designer indicates the uncovered ASpec more accurately than presenting F_2.

Since a temporal property can be decomposed in many ways, the task of determining the weakest set of architectural sub-properties that can accurately characterize the coverage gap in a meaningful way is a non-trivial task. We have solved the problem for a restricted, but rich fragment of LTL. More detailed examples involving more than one component modules has been presented in [2].

3. Formalization

The primary coverage question is to determine whether the ISpec covers the ASpec. Not surprisingly, the answer to this question follows from testing an appropriate logical implication. The formal proofs of all results presented in this paper are given in [2].

Theorem 1 *The ISpec of a device D covers the ASpec iff $\mathcal{I}^D \Rightarrow \mathcal{A}^D$ is a tautology, where $\mathcal{I}^D$ denotes the ISpec of D and $\mathcal{A}^D$ denotes the ASpec of D.*

Definition 1 [Strong and weak properties]
A property $\mathcal{F}_1$ is stronger than a property $\mathcal{F}_2$ iff $\mathcal{F}_1 \Rightarrow \mathcal{F}_2$ and $\mathcal{F}_2 \not\Rightarrow \mathcal{F}_1$. We also say that $\mathcal{F}_2$ is weaker than $\mathcal{F}_1$.

Definition 2 [Uncovered ASpec]
An uncovered ASpec is a property A_H over $\mathcal{AP}_A(D)$, such that $(\mathcal{I}^D \wedge A_H) \Rightarrow \mathcal{A}^D$, and there exists no property A'_H over $\mathcal{AP}_A(D)$ such that A'_H is weaker than A_H and $(\mathcal{I}^D \wedge A'_H) \Rightarrow \mathcal{A}^D$. In other words, we find the weakest ASpec property that suffices to close the coverage hole.

4. Finding the Uncovered ASpec

Let us consider the following fragment of Linear Temporal Logic [1]. The syntax of the logic is defined over a set of atomic propositions $\mathcal{AP}$. The proposed fragment is defined by the following grammar:

Q: G(φ)
φ: True | False | $p \in \mathcal{AP}$
| $\neg\varphi$ | $\varphi \wedge \varphi$ | $\varphi \vee \varphi$ | $\varphi \Rightarrow \varphi$ | X φ

The semantics of these operators are standard [1]. We have found that the AMBA AHB protocol and the PCI Bus protocol properties can be modeled in this fragment of LTL, which shows that the above fragment is in fact quite rich in practice.

Theorem 2 *If the ASpec and ISpec consists of Boolean formulas, then the uncovered ASpec, A_H, is unique and given by $UABS(I_H, Z)$, where I_H is $\mathcal{A}^D \vee \neg\mathcal{I}^D$, $Z = \mathcal{AP}_I(D) - \mathcal{AP}_A(D)$, and $UABS(I_H, Z)$ is the universal abstraction of I_H with the set of propositions in Z, where Z contains the set of atomic propositions that appear in the ISpec but not in the ASpec.*

Lemma 1 *If the ASpec, $\mathcal{A}^D$, consists of Boolean properties only, and A_H is the uncovered ASpec with respect to the ISPec, $\mathcal{I}^D$, then the uncovered ASpec of $G[\mathcal{A}^D]$ with respect to $G[\mathcal{I}^D]$ is unique and given by $G[A_H]$.*

Theorem 2 and Lemma 1 solve the coverage problem for the above fragment of LTL without the X operator. For properties having the X operator, we translate the property into the following normal form.

Definition 3 [X-Normal form]
A property specified using the proposed fragment of LTL is in X-normal form iff each X operator encloses a single proposition from $\mathcal{AP}$.

Lemma 2 *Every property in the proposed fragment of LTL can be specified in X-normal form.*

We have developed a methodology of transforming this logic to it's Boolean equivalent and then returning the Boolean coverage hole into the uncovered ASpec in our logic. The details of this methodology is given in [2].

References

[1] Clarke, E.M., Grumberg, O., and Peled, D.A., *Model Checking*, MIT Press, 2000.

[2] Das, S., Basu, P., *et al*. Detailed version of this paper. Email: {sayantan,pbasu}@cse.iitkgp.ernet.in