Chapter 6

Conclusion

In this thesis, an approach to handle the combinational permutation independent Boolean comparison problem is presented, i.e., the problem whether two Boolean functions f and g are equivalent independent of the permutation of their input variables. The approach concentrates on establishing a possible correspondence for equivalence between the input variables of f and g. This problem is $\mathcal{NP}$–hard. So heuristic solutions are necessary. The approach introduced here uses signatures for an input variable of a Boolean function f to handle the problem. In this context, a signature is a special information about an input variable of a Boolean function f which is independent of the permutation of all input variables of f. The data structure which is used to represent Boolean functions is the reduced ordered binary decision diagram (ROBDD).

The following two observations could be made:

1. *Signatures are especially well–suited to uniquely identify input variables independent of permutation.*

   Three classes of signature functions are presented which were proven to work very efficiently. In fact, in all cases were it was possible to uniquely identify an input variable in our large set of benchmarks these signature functions were able to do it. Furthermore, in order to compute several of these signature functions just one ROBDD describing the Boolean function is needed, and of course, the computation does not depend on the variable ordering of the ROBDD. Thus as long as we can construct an ROBDD for the function with any variable ordering, we can apply these signature functions. This is a very modest requirement, since an inability to construct the ROBDD for any variable ordering would preclude verification with the help of ROBDDs even if you knew the variable correspondence.

2. *The only limitation for signatures to be able to uniquely identify input variables are special symmetries, that we call $\mathcal{G}$–symmetries.*

   At first the limitations of using signatures to tackle the combinational permutation equivalence problem are examined by presenting basic results which identify exactly what these limitations are. The property of $\mathcal{G}$–symmetry of Boolean functions is investigated, and a universal signature function is introduced which dominates all other signature functions in the following sense: if there is any signature function which can distinguish between two input variables of a Boolean function, then the universal signatures of these two variables must be different as well. In this sense, the universal signature function is the strongest signature function which can be constructed. Then the existence of a universal signature could be proven and so a central theoretical result of this thesis: if any two input variables of a Boolean function f have the same universal signature, then there is a permutation $\pi \in \mathcal{P}_n \setminus \mathbf{1}$ of the input variables of f such that applying this permutation to the inputs of f does not change this function. In other words, the only limitation for signatures to be able to uniquely identify input variables are $\mathcal{G}$–symmetries.

Next, new kinds of symmetry classes (i.e., special $\mathcal{G}$–symmetries) are identified that help in finding a correspondence between the input variables of two Boolean functions being compared.

For our large set of benchmark circuits, the CPU–times necessary to establish such a unique correspondence are very promising. Thus, in addition to providing theoretical insight, the algorithms presented have direct practical impact for the complete solution of the permutation independent Boolean comparison problem.

For years, the permutation equivalence problem has been worked on by several other authors as well [9, 10, 21, 25, 29, 31, 33]. Considering all the approaches for handling the permutation equivalence problem that were proposed in the last few years we can say the following. It is not possible to say that one of these approaches presents the best method to handle the problem. Each of these methods will work well for a special class of practical circuits. However, except for the approach presented in this thesis, none of them takes general kinds of symmetry (that we call $\mathcal{G}$–symmetry) into account. That may be acceptable for application in technology mapping when just a small number of inputs is involved. There, it may be feasible to try all correspondence possibilities established after applying different signature functions or try using the method introduced in [31]. However, when permutation independent Boolean comparison has to be used for functions with a large number of inputs, as in formal logic verification, it is definitely necessary to take $\mathcal{G}$–symmetries into consideration. This can be underlined by the theoretical investigations with respect to the limits of signatures and by the experiments on the large set of benchmarks discussed in this thesis. So, applying the methods that use the knowlegde about $\mathcal{G}$–symmetries significantly sets the work of this thesis apart from the other approaches.

Furthermore, it is shown that these methods can be used in order to handle other problems of circuit design and verification as well. Here, it is demonstrated how the methods can be easily extended and applied to handle one problem in sequential logic verification. That is the problem of establishing the unknown correspondence between the latch variables of two sequential circuits with the same state encoding. We call it the latch permutation equivalence problem. A solution of this problem can be used to verify the combinational equivalence of two sequential logic circuits that have the same state encoding, but the correspondence between the latch variables is not known.

Experiments have shown that as long as there are no latch symmetries, signatures can be used to establish a unique possible correspondence between the latches. The CPU times necessary to establish such a unique correspondence are very promising.

Moreover, this method can be easily extended to identify equivalent latches in a circuit (see Section 5.4). Thus we believe that it is especially suited to be added in sequential verification tools that use product machine traversals. Here, computations can be made more efficient by exploiting combinationally equivalent state variables (see for example, [13, 36]).

In our opinion, the methods presented here to tackle the latch correspondence problem can easily be integrated with existing verification methods and so significantly improve the ability of these techniques to handle sequential circuits with the same state encoding. This has direct practical impact in sequential logic verification because it enlarges the class of sequential circuits for which verification is feasible (see also [13]).

Finally, there are several ideas for future projects related to the work reported in this thesis.

The thesis shows that it is extremely difficult to handle $\mathcal{G}$–symmetry in general. Furthermore, it demonstrates the practical importance of $\mathcal{G}$–symmetry of Boolean functions for logic synthesis and verification. So it would be useful to investigate this further.

However, there are other important fields of investigation. One of these is to check other data structures, i.e., other kinds of decision diagrams (for example OKFDDs [2]), circuit descriptions on the gate level, etc., for their ability to be used in applying these signature–based methods for solving the permutation problem. The advantage of this is that it would enlarge the class of combinational and sequential circuits for that the permutation problem can be efficiently solved.

Another important area is the application of the signature–based methods to incompletely–specified Boolean functions. With the extension of these methods to functions with DC's, several improvements in different areas of circuit design and verification would be possible. In technology mapping, it would enlarge the degrees of freedom for efficient mappings of Boolean functions to a certain library. Some preliminary investigations on this issue were made in [16].

Moreover, it would be interesting to extend the approach to handle the combinational equivalence problem for unknown input *and* unkown output correspondence of two Boolean functions with more then one output. Here, the application of signature functions to the output variables of a Boolean function is straightforward: satisfy count signatures and breakup signatures could also be used to identify output functions. Furthermore, the permutation independently ordered signature vector of the input variables of an output function could be used to identify this output function independent of permutation. The influence of this extension regarding to $\mathcal{G}$–symmetry and the heuristics to handle special kinds of this symmetry has to be investigated.