論文の内容の要旨

論文題目　Debug Support for VLSI Designs Based on Error Trace Analysis
（エラートレース解析に基づく VLSI 設計のデバッグ支援）

氏　名:　李　蓮福

In the last decades, size and complexity of VLSI designs have remarkably increased with the advance of process technology. Accordingly, design debugging cost has increased greatly, and it is becoming one of the largest bottlenecks in a design cycle. Especially, post-silicon debugging, which targets the errors detected by running the fabricated chip, is generally much harder than pre-silicon debugging. It is because not only the observability and the controllability of internal signal values are very low but also the execution cycle of a fabricated chip is generally very long. Now, post-silicon debugging has become a major time consuming step that imposes a significant impact on the design cycle, consuming 35 % of the entire development period. Therefore, efficient debugging supports are strongly required to improve the productivity of VLSI designs.

In practice, hardware design debug has been mainly relying on manual task. That is, designers have to identify the root cause of the target error by analyzing the simulation or execution traces depending on their knowledge of the design, specific expertise, and experience. In trace-based manual debugging, the debugging effort is proportional to the amount of trace data. Furthermore, in post-silicon debugging, only a part of the entire traces can be provided and the execution cycles are typically extremely long, which makes it much harder to analyze the trace. A considerable number of signal values is contained in such traces, which are not actually relevant to the target error. Therefore, improving the efficiency of error trace analysis process is the key for efficient debugging.

To improve the efficiency of the manual task in trace-base debugging, three methods are proposed in this thesis; (1) The dynamic signal sequence slicing method, (2) The I/O sequence mapping method between a high-level design and a low-level design, and (3) A framework for debugging low-level designs using a high-level design.

First, in this thesis, a method that automatically extracts the signal values relevant to the target erroneous value from the original error traces is proposed. First, a new 1-bit variable

named d-tag is proposed which is to be generated for every signal such as input signals, data registers, and output signals. The value of d-tag of a signal *s* at each cycle is set to represent whether the value of *s* at that cycle is relevant to a specified signal value at another cycle. Formulas computing d-tags for signals in a design are established from the Finite State Machine with Data-path (FSMD) representation of the design. Since each d-tag is a 1-bit value and its computation is conducted by a pre-defined Boolean formula of simple construction, the method can be achieved with a small computation cost. By applying this method, we become able to identify which signal values from the erroneous trace are actually relevant to the target error in trace-based debugging. This relieves us of the effort to be taken for examining and analyzing those values. Especially, it is even more effective in post-silicon debugging, where there are a significant number of such irrelevant signal values in error traces. The proposed dynamic signal sequence slicing method can be implemented either by software or on-chip hardware. The functionality of the proposed slicing method is demonstrated using several design examples. As a result, we observed that the numbers of the extracted signal values was significantly smaller than the size of the original sequences. Moreover, implementation of the proposed method as an on-chip hardware circuit is also demonstrated. As a result, we confirmed that the area overhead due to the additional circuitry for performing the proposed dynamic signal sequence slicing method was quiet practical, e.g., 1.95 % for the IDCT design.

Meanwhile, recently, designs of higher abstraction levels than conventional RTL (Register Transfer Level), such as behavior-level or system-level are increasingly employed as a starting point of a design. Such high-level designs provide higher readability of design description and are easy to understand the behavior compared with lower-level designs. Accordingly, trace-analysis of the erroneous behavior observed in the simulation of such high-level designs are much efficient than that of low-level design simulation or chip execution. However they are not utilized in debugging of a low-level design including a fabricated chip because of the inherent difference of interfaces between a high-level design and a low-level design.

Focusing on the above observation, an automatic mapping method between I/O sequences of a high-level design and a low-level design including a fabricated chip is proposed. The ultimate purpose is to take the advantage of high-level design by utilizing the high-level design simulation in debugging the corresponding low-level design. To achieve this, first, a formal definition of I/O sequence mapping and the relevant notions are provided, such as *port relation*, *timing relation*, and *valid value sequence*. Based on those notions, the

mapping method is achieved through the two steps: (1) Extract a valid value sequence for each signal from the I/O sequences of a chip execution (2) Generate high-level I/O sequences by mapping from the valid value sequences. By applying this method, we become able to analyze the erroneous behavior observed in the low-level design simulation or chip execution by high-level design simulation, which provides higher efficiency in debugging. The effectiveness of the proposed method is demonstrated by experimental results using several design examples. At that time, the runtime for conducting the mapping process were practical. As a result, the numbers of cycles and I/O events for the the high-level designs simulation with the I/O sequences obtained by the mapping method were significantly reduced compared to the conventional RTL simulation. Furthermore, an on-chip implementation of the method to deal with such I/O sequences during very long execution cycles is presented, and the experimental results are demonstrated. As a result, the area overhead of each design due to the additional circuitry for performing proposed mapping method was also quiet small, e.g., 0.01% for JPEC encoder design.

Finally, a low-level design debugging framework utilizing high-level design is proposed, for further effectiveness. It presents an effective way of utilizing the dynamic signal sequence slicing method and the I/O sequence mapping method presented above in debugging. The goal of the framework is to utilize high-level design simulation with the signal sequences that are extracted from the original long error trace by dynamic signal sequence slicing method. Moreover, methods to extract only the error-relevant portions from the high-level design with the degree suspicious to be buggy portions are proposed to get more debugging efficiency. The extraction method of error-relevant portions is achieved by utilizing the dynamic program slicing method for a high-level design. Also, a distance metric to evaluate the suspicious degree to be a buggy portion is proposed. The experimental results show that we can have the number of design portions reduced that we have to analyze when debugging, by applying the framework.

The experiments conducted in this thesis confirm that the proposed methods contribute to improving the efficiency of debugging large size and complicated designs effectively.