A Selective Error Data Capture Method using On-Chip DRAM for Silicon Debug of Multi-core Design

Hyunggyo Oh, Heetae Kim, Jaeil Lim, and Sungho Kang
Yonsei University, Seoul, Korea
Email: [kyob508, kht2161, limji87]@soc.yonsei.ac.kr, shkang@yonsei.ac.kr

Abstract—In multi-core designs, the time overhead for the post-silicon debug is a main challenge because of a large number of cores under debug and the limited resource of design for debug. To overcome this challenge, we propose a selective error data capture method using on-chip DRAM. The key idea is that it is not necessary to capture error-free data of each core. First, the error interval matrix is generated to detect the erroneous intervals of each core by using a multiple-input signature register. And then, the erroneous data capture sequence is used to minimize the number of debug sessions using the debug scheduling algorithm. The experimental results show significant debug time reduction with a negligible hardware overhead compared to the previous work.

Keywords—On-chip DRAM, post-silicon debug, multi-core design, multiple-input signature register (MISR).

I. INTRODUCTION

Semiconductor technology has been developed and has allowed for the integration of a large number of cores into a single system-on-chip (SoC). Because of the growing complexity of a SoC, it is more difficult to ensure that the design is error-free. In addition, there are some errors which escape from pre-silicon verification and manufacturing test and these errors adversely affect the time-to-market requirements. Therefore, post-silicon debug and validation have been an important role of the circuit implementation flow [1], [2], [3]. The trace buffer-based debug method is commonly used to observe real-time signals without stopping the functional operation. Since the real-time trace signals are captured in the trace buffer and the size of the trace buffer results in design for debug (DfD) hardware overhead, the main challenge of the trace buffer-based debug is the limited observability. In order to overcome this limitation, the DRAM-based debug method has been introduced. In [4], a massive signal tracing method using on-chip DRAM which can be integrated in a SoC or through-silicon via (TSV)-based 3D-ICs has been introduced. This method requires only one debug session when debugging one core and it is a significant debug time reduction compared to the trace buffer-based method. However, there is no consideration of the debug case of multi-core design in this method. That is, this method still requires as many debug sessions as there are cores under debug (CUDs).

In this paper, a selective error data capture method using on-chip DRAM for the post-silicon debug of multi-core design is proposed. First, a multiple-input signature register (MISR) is used to detect the error interval of each core. With this information, the debug data capture sequence is determined to minimize the number of debug sessions. This method supports the post-silicon debug of the multi-core designs using on-chip DRAM in order to reduce the debug time.

II. PROPOSED DEBUG SCHEME

In this section, the overview of the proposed debug flow, the proposed DfD architecture, details of each debug step and the consideration of the proposed DRAM-based debug method are discussed. The proposed method consists of two debug steps, which are error interval detection and selective error data capture. First, a MISR set is used to detect the error interval of each core and the error interval matrix is generated. With this matrix, only the erroneous data interval of each core can be captured selectively. To perform this selective data capture, erroneous data capture sequence (EDCS) is generated to minimize the number of debug sessions. The simple example is illustrated in Fig. 1 where colored boxes indicate the error intervals of each core. Unlike the sequential debugging of the previous method, multiple cores can be debugged using EDCS at the same time and the number of debug sessions can be reduced. In this example, debugging core 1, 2 and 3 can be performed in EDCS 1. As a result, the proposed method provides a significant reduction in the debug time for multiple cores when compared to that for the previous method.

The overview of the proposed DfD architecture is described in Fig. 2. First, the error intervals are detected by the golden MISR signatures. The golden signatures are generated and...
loaded in the debug configuration step. With the results of interval detection, debug scheduling is performed as generating the set of EDCSs. After that, error data is selectively captured in the trace buffer set with the set of EDCSs, transferred to the shadow buffer set and stored in the DRAM. Distributed trace buffers are generally used in multi-core designs [2], [3] and some trace buffers are reused as the shadow trace buffer in the proposed architecture.

As discussed before, DRAM-based debug method requires the shadow buffers to hold the captured debug data until they are transferred to the DRAM because of the DRAM operation. In addition, the capture information such as data sequence and time stamp is also stored in the DRAM. The control logic configures the trigger points and provides control signals for communication fabrics and read/write DRAM access operation.

III. EXPERIMENTAL RESULTS

This section describes the experimental results in terms of DRAM usage and debug time in order to illustrate how the proposed method improves the previous work [4] in multicore debug cases. The experimental results are presented for an ARM-based processor design. Each debug module is designed as a Verilog RTL model and synthesized using a 130 nm CMOS library to estimate the hardware area size. Faults are randomly injected into the circuits and a 32-bit data bus is assumed as the debug data to be observed. It should be noted that the hardware area overhead is negligible because the proposed method re-uses DfD modules which are used in [4] and requires only some MISRs and comparators.

Fig. 3 shows the experimental results of the debug time with different trace buffer depth size for the debug cases in which the observation window is 512 K cycles, single trace/shadow buffer set and the error rate is 0.179%. SE indicates the segmentation ratio of the trace buffer in the proposed method. It should be noted that the trace buffer size and SE can be limited by the DRAM specification. The proposed method can reduce the debug time with the selective error data capture and the total debug time is reduced when the trace buffer size is small. This is because trace buffer size (M) indicates the error detection interval size and the shorter interval can detect the error data minutely. Although the debug time reduction ratio is reduced as SE increases, the debug time is significantly reduced compared to [4].

IV. CONCLUSION

The main challenge of post silicon debug is to maximize observability of internal states and DRAM-based debug method can overcome this challenge by exploiting the on-chip DRAM such as embedded DRAM or 3D DRAM. In this paper, a selective error data capture method for silicon debug of the multi-core design is proposed to reduce the debug time. An error interval detection is performed by MISR signatures and a selective error data capture is scheduled by set of EDCSs. By using two debug steps, the debug process is performed with significant debug time reduction if the consideration of the DRAM specification is satisfied.

ACKNOWLEDGMENT

This work supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2015R1A2A1A13001751)

REFERENCES