5A-3: Hardware Debugging Method Based on Signal Transitions and Transactions

Nobuyuki Ohba and Kohji Takano
IBM Research, Tokyo Research Laboratory, IBM Japan Ltd.
Contents

- Background and problems
- Idea and features
- Hardware implementation
- User interface - visualizer
- Summary and future work
- Visualizer demonstration
Background

- Hardware design becomes bigger and deeper
- Debugging becomes more complicated and time-consuming
  - Bugs dive into “deep sea,” million gates of ASIC/SoC
  - It takes long time, hours, days, and even weeks, to test and verify the system
- Logic analyzer is a powerful tool, but...
  - Because of limited amount of memory, logic analyzer covers only a partial period of long run
  - Even if all the transitions are captured, it is difficult to trace timing charts by “human eyes”
  - “Try and error” various trigger conditions based on intuition and six sense
Problem examples

I ran a test program on the debugging board and got an error. I ran the program again. No error. I ran the program ten more times, but still no error. What was the error?

I am using a logic analyzer to trace an error. I have no idea on how I could set the trigger condition.

I am designing a new IP core. I had an error when I got my IP connected to Design X from Company Y. I only have a simple data sheet for Design X. My IP should work. Why not?
Transition and transaction tracer

- Probe and capture signals for a long time, **hours, days, or even weeks**
- Record all transition patterns in a state-transition diagram format using real-time compression and pattern matching
- **Automatically** generate trigger using newly and/or rarely detected transition
- **Visualize** the transitions to help designers identify the bug

![Diagram of DUT, Tracer, and Visualizer](image)

*Probed signals, Real-time compression, Real-time pattern matching, State transition diagrams, Record all transitions, Time, Trigger, New*
Generate a trigger when a new transition is detected. Rare transitions often give a good hit for debugging. They are often unexpected or not verified.
Visualizer

- Visualize signal transitions to help engineers tackle problems
  - Matrix view
    - Show transition counts between back-to-back states
    - Make it easy to find rare transitions
  - State transition diagram view
    - Show all the transition patterns between idle states
    - Colored by attributes (new, frequent, rare, and user selected transitions)
    - Fully automatic depiction
  - Timing chart view
Each cell shows transition counts between back to back states
- Cells on Column A and Row B shows the number of captured transitions from State A to State B

Trigger is generated by
- Predefined condition
- Newly detected transition

Blue colored cells are “Transition counts < N”
Rare transitions often give a hint for debugging; they tend to be unexpected or not verified
State transition diagram view

Blue Line: Transitions that contain the user selected node

Green Line: Path selected by user

Red Line: New Transition

Timing chart view of the selected transaction
Prototype implementation using FPGA

- **Tracer**
  - Made prototype using Xilinx Virtex-II FPGA (XC2V4000)
  - Runs at up to 100MHz
  - Requires 10k to 100k ASIC equivalent gates (depending on the number of signals and trace depth)

- **Visualizer**
  - Windows GUI
  - See the demonstration
Problem examples

I ran a test program on the debugging board and got an error. I ran the program again. No error. I ran the program ten more times, but still no error. What was the error?

I am using a logic analyzer to trace an error. I have no idea on how I could set the trigger condition.

I am designing a new IP core. I had an error when I get my IP connected to Design X from Company Y. I only have a simple data sheet for Design X. My IP should work. Why not?
Problem examples - solution

I ran a test program on the debugging board and got no error detected when ran the program several times. What was the transition from state 1 to 3 caused the trouble. The difference gave me a good hint.
Problem examples - solution

I am using a logic analyzer to trace an error. I have no idea on how I could set the trigger. After 3-hour test run, a new transition was detected. It triggers logic analyzer. That helped me pinpoint the bug.
I am designing a new core. I had an engineer who got my IP connectivity for Design X from C. I only have a single sheet for Design X, which should work. When I did not expect X would happen in C.

Specification of Design X was unclear.
Summary and future work

- **New hardware debugging tool – Transition and Transaction Tracer**
  - Probe and record signals for a long time (hours, days, weeks) without intermittence
    - Real-time data compression
    - Real-time transition pattern matching
  - Automatically generate trigger signal
    - Trigger by newly and/or rarely detected transition
  - Visualize signal transitions in matrix, timing chart, and state transition diagram views

- **Future work**
  - Embed the tracer into ASIC
  - Use for software debugging
State Diagram View Demonstration

Please see this demonstration in the full-screen slideshow mode.

To quit the demonstration, click the mouse button.