**Data-Triggered Breakpoint for In-Circuit Debug without Re-implementation**

* Yutaka Tamiya  
* Yoshinori Tomita  
* Toshiyuki Ichiba  
* Kaoru Kawamura  

Fujitsu Laboratories Ltd., Japan

---

**Implemented with Simple HW**  
Linearity of CRC  
(Immediate Redundancy Check)  

**Realize Fast & Flexible Debug**  
At-Speed & Realtime  
With Small Area Overhead  
No Re-implementation

---

**Background**  
- Problems of Post-Silicon Debug  
- Logical Bugs remain even after Pre-Silicon.  
- Impractical Simulation with exhaustive Test Patterns.  
- Needs much Efforts to Detect & Fix Bugs.  
- Poor Controllability & Observability  
- Takes Long TAT with In-Circuit Logic Analyzer: ChipScope, SignalTap II, etc.  
- HW re-implementation required on changing Breakpoints.  

**Growth of Hardware: Size, Speed, Func.**  
- Ethernet=40~100Gbps, PCIe=5~8Gbps/Lane  
- Almost all systems consist of HW & SW

---

**Algorithms to Detect Data Sequences**  
- Divide Data Seq, S (arbitrary length of n) into  
  Sub-Seq's: S1, S2, ..., Sr (Len of Sub-Seq's: n1, n2, ..., nr)  
  S = S1 & S2 & ... & Sr  
- S1: The 1st Sub-Sequence included in the 1st Word.  
  → ①CRC Difference Detection  
- Sk's: The 2nd and Latter Sub-Sequence's (k=2,3, ...,r) with arbitrary Lengths n2, ..., nr.  
  → ②Absolute CRC Detection.

---

**CRC Diff Detection for S1**  
Try to detect S1 in every word:

**Abs CRC Detection for S2**  
S2, following S1, can be detected by:  

---

**Rapid Debug Cycle**

---

**Results of Resource Usage**  
- Synthesized with Xilinx Kintex7 @Clock 250MHz by Xilinx ISE Tool.  
- Small Area Overheads with 4 or less of CRC Width.  
- Data Seq = 595 MB JPEG stream (as Random Data)

---

**Results of False Positives**  
- False Positives can be reduced by CRC-Width, # of Sub-Sequence, and Word Width.  
- Data Seq = 595 MB JPEG stream (as Random Data)  
- A 64-byte “SOS” sequence to be detected

---

**Future Work**  
- Investigate more complex Breakpoint Conditions w/ closer Collaboration between Debugger(SW) & Debug Module(HW)  
- Combine our breakpoints w/ other Observability Technique and Realize a practical Debug Tool.