



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

Nobuyuki Ohba and Kohji Takano

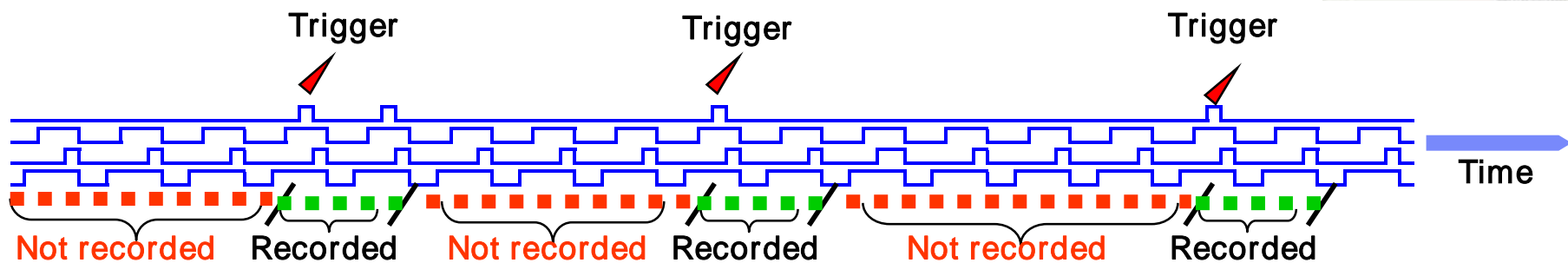
IBM Research, Tokyo Research Laboratory, IBM Japan Ltd.

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?

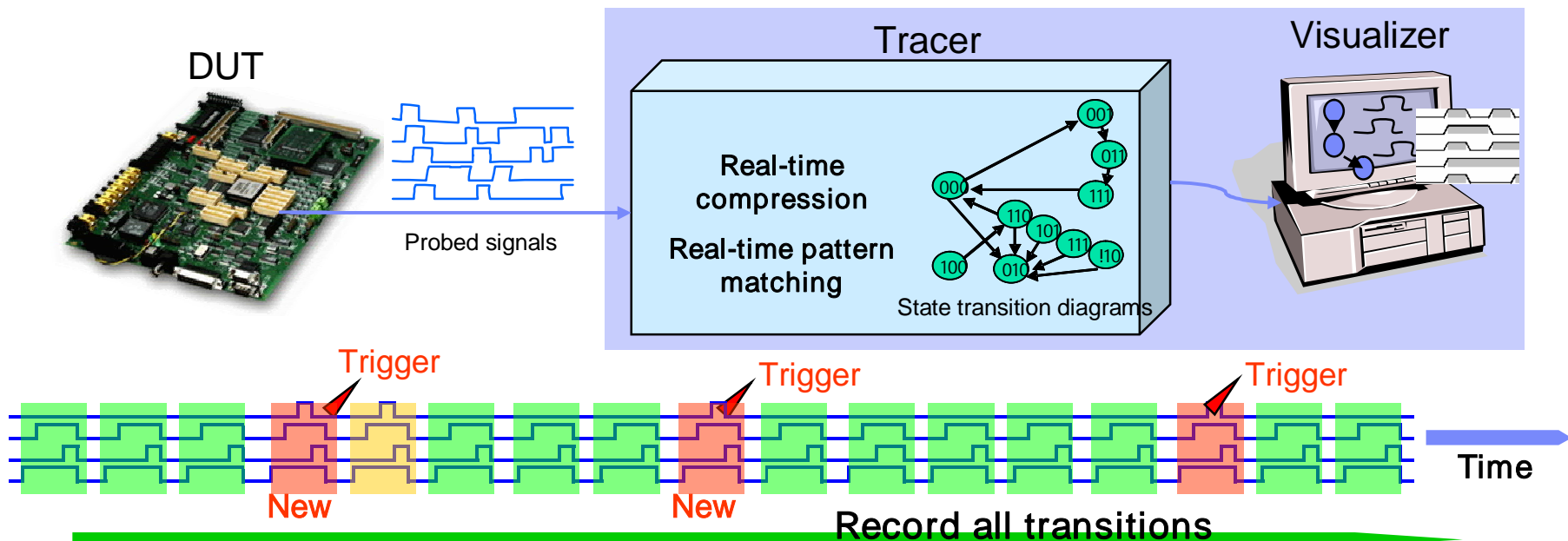
I am using a logic analyzer to trace an error. I have no idea on how I could set the trigger condition.

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?

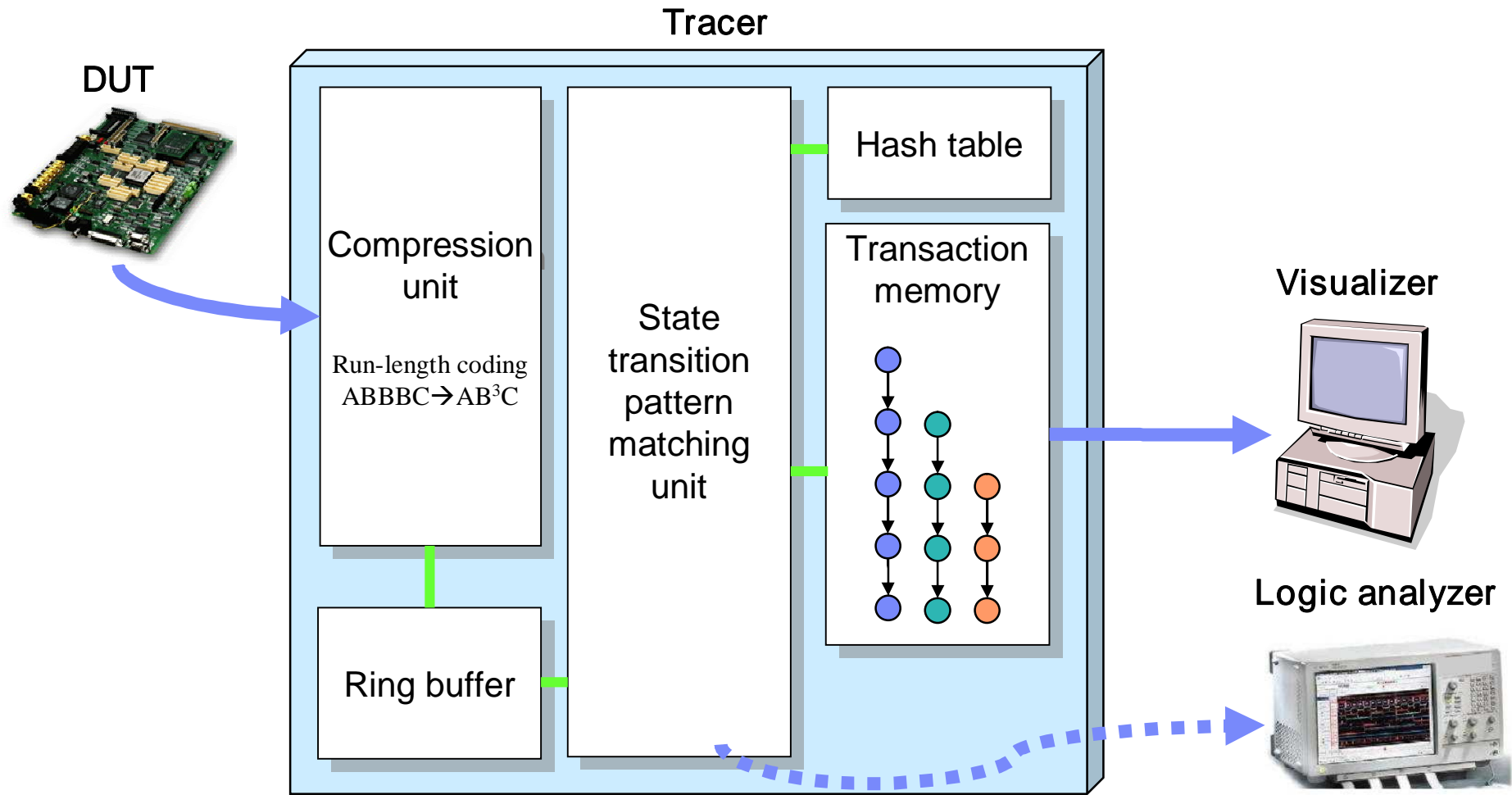


Transition and transaction tracer

- 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



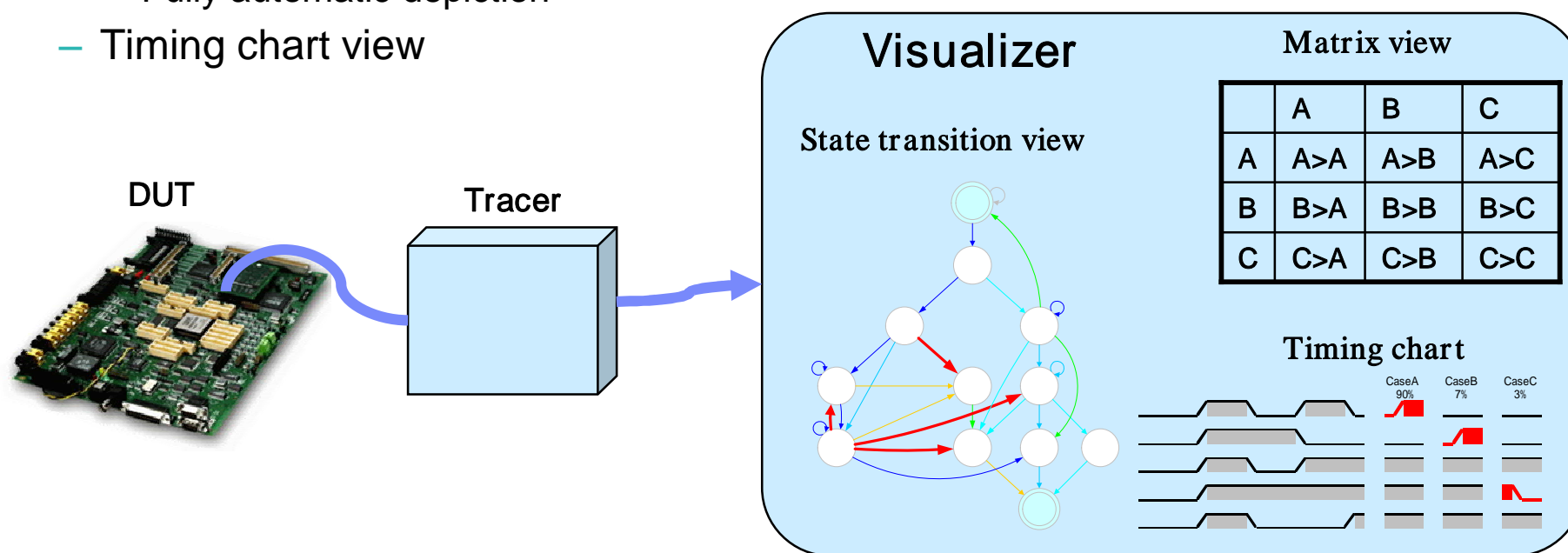
Tracer



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



Matrix view

N

	0001000000	0100000000	0101000000	0101100000	0111100000	1000000000	1001000000	1100000000	1101000000	1101100000	1111100000
0001000000	000000001e9b						0000000005ad				
0100000000								00000000076b			
0101000000	000000000211	000000000075	0000000019f5								
0101100000	0000000002af	000000000001	000000000366								
0111100000				000000000516					000002d670de		
1000000000											00000015768
1001000000											0000020b7667
1100000000											00000008e741
1101000000						00000015768	00000073b00d	00000008e80b	000000a121ed		
1101100000							00000257d0b6		0000007dee3c	00000002474a	0000000c110
1111100000					000002d67610						000110216d2f

- 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

```

m_display_list_elements:11 pDoc->jtag.local_l1cam_size:24
m_display_list_elements:11 pDoc->jtag.local_l1cam_size:24
m_display_list_elements:11 pDoc->jtag.local_l1cam_size:24
m_display_list_elements:11 pDoc->jtag.local_l1cam_size:24
m_display_list_elements:11 pDoc->jtag.local_l1cam_size:24
m_display_list_elements:11 pDoc->jtag.local_l1cam_size:24
    
```


State transition diagram view

The screenshot displays the EmbeddedTranomon interface with several key components:

- State Transition Diagram:** A central diagram showing states (nodes) and transitions (edges). Nodes include DLE_IN, 1e0, 160, 040, 140, 100, 360, 340, 200, 300, 240, and IDLE_OUT. A green path is highlighted from DLE_IN through 1e0, 160, 040, 140, 100, 300, 200, 240, and IDLE_OUT. A red arrow points to a new transition from 040 to 240.
- Transaction List:** A table titled "Display all transactions" with columns for Length, Iteration, Serial, and data fields [0] through [6]. Transaction 22 is selected.
- Timing Chart:** A dialog box titled "Test transaction no 1" showing waveforms for signals: low, DEVSELn, STOPn, IRDYn, TRYDn, and FRAMEn.

Annotations and callouts:

- 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**

Transaction List Data:

Length	Iteration	Serial	[0]	[1]	[2]	[3]	[4]	[5]	[6]
19	000000003	33	Null	1e0	Null	160	Null	140	Null
14	000000003	31	Null	1e0	Null	160	Null	140	Null
14	000000003	42	Null	1e0	Null	160	Null	140	Null
26	000000002	36	Null	1e0	Null	160	Null	140	Null
19	000000001	41	Null	1e0	Null	160	Null	140	Null
12	000000001	43	Null	1e0	Null	160	Null	140	Null
34	000000000	46	Null	1e0	Null	160	Null	140	Null
27	000000000	35	Null	1e0	Null	160	Null	140	Null
22	000000000	5	Null	1e0	Null	160	Null	140	Null
22	000000000	40	Null	1e0	Null	160	Null	140	Null
18	000000000	48	Null	1e0	Null	160	Null	140	Null
15	000000000	34	Null	1e0	Null	160	Null	140	Null
10	000000000	47	Null	1e0	Null	160	Null	040	Null
10	000000000	44	Null	1e0	Null	160	Null	140	Null

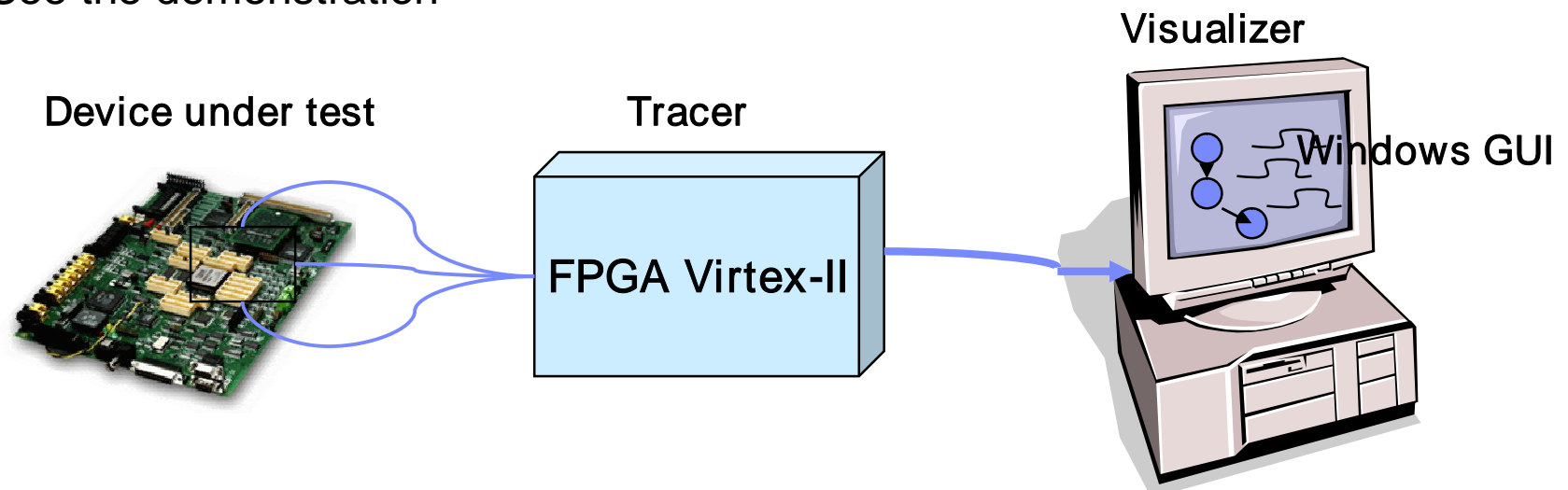
Prototype implementation using FPGA

■ 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.

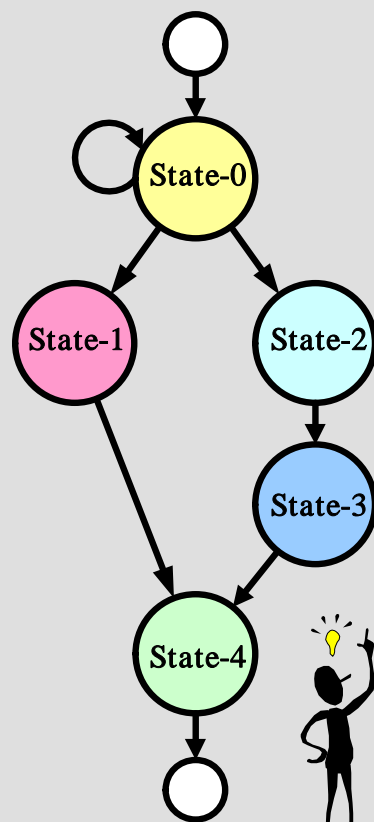
I am designing a new IP core. I had an error when I get my IP connected to Design X from Company Y. I only have a simple data sheet for Design X. My IP should work. Why not?



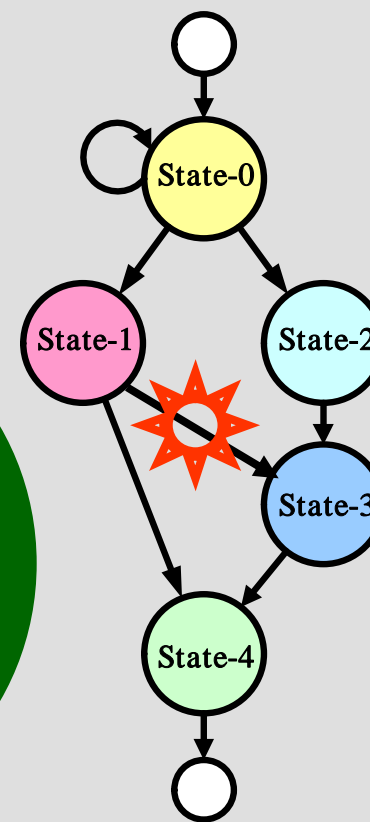
Problem examples - solution

I ran a test program on the debugging board and got program ran the times. When

No error detected



Error detected

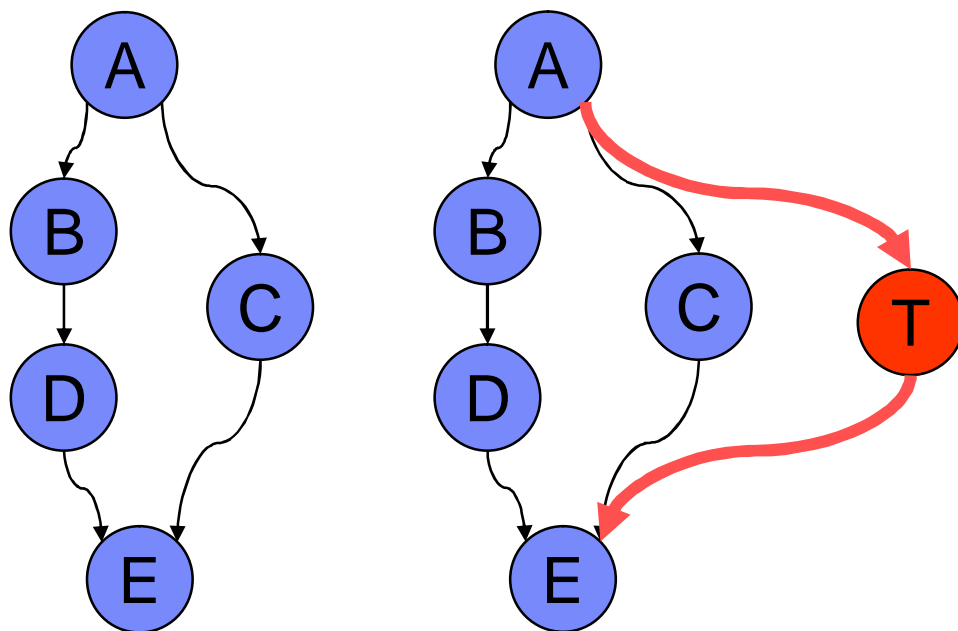


Compare

Transition from state 1 to 3 caused the trouble.

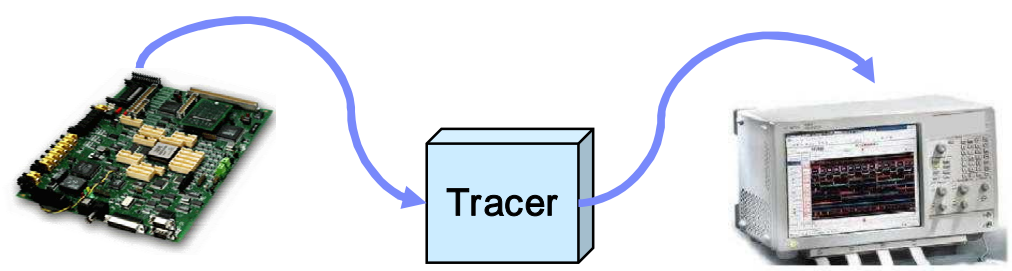
The difference gave me a good hint.

Problem examples - solution

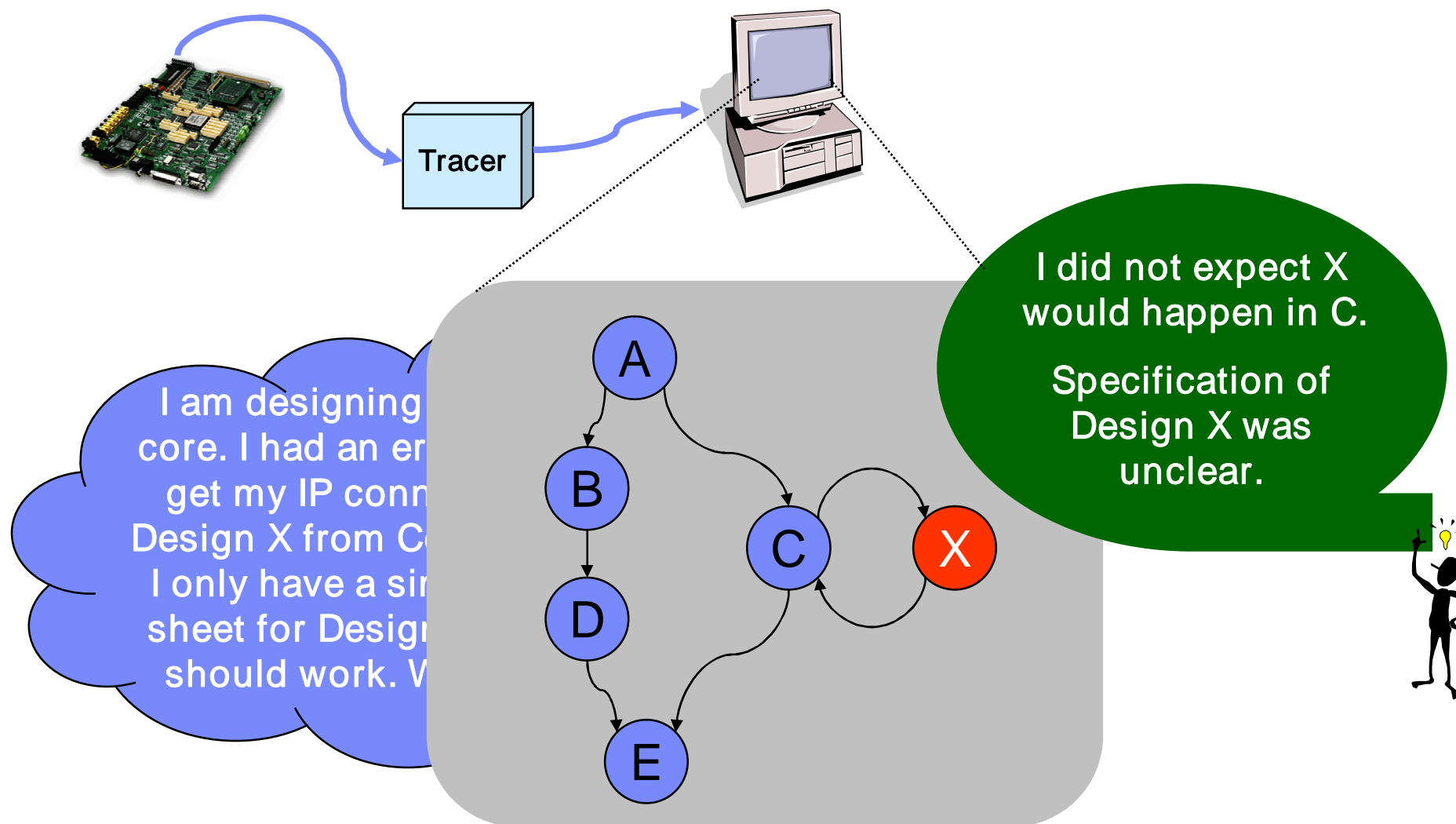


I am using a logic analyzer to trace an error. I have no idea on how I could get the trigger.

After 3-hour test run, a new transition was detected. It triggers logic analyzer. That helped me pinpoint the bug.



Problem examples - solution



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.