Hazard Detection Tool in Asynchronous Finite State Machines Transition Logic

life.augmented

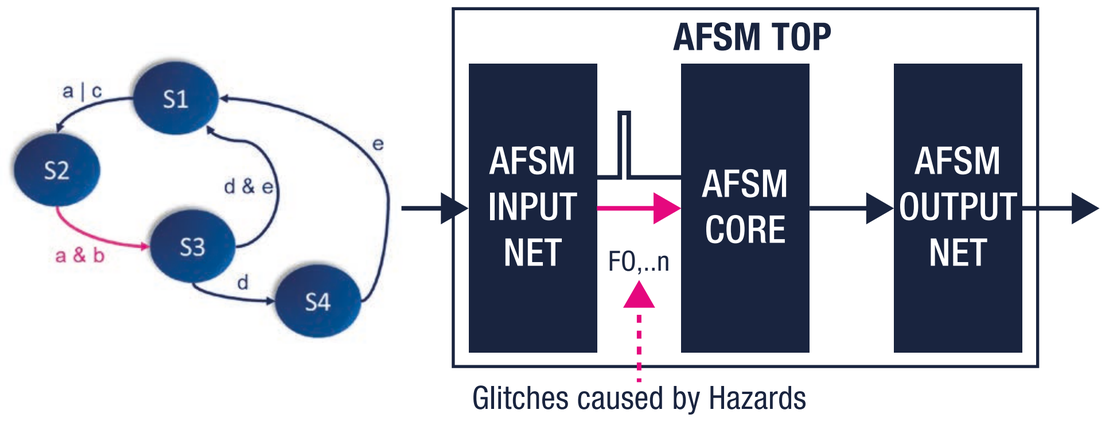
By Roberta Priolo, Francesco Battini, Enea Dimroci from STMicroelectronics
eMail: roberta.priolo@st.com, francesco.battini@st.com, enea.dimroci@st.com

1. INTRODUCTION

The presence of hazards in asynchronous circuits is especially critical.

In Asynchronous Finite State Machines (AFSMs), glitches can lead to unwanted state transitions.

The AFSM generation flow must guarantee that the Hazards inside AFSM logic are detected and appropriately solved.



2. PROBLEM SOLVED

A Hazard Detection Tool has been implemented to detect the hazards in the AFSM Arc Transitions Logic.

It focuses on the ones that cause rising glitches because they can unpredictably enable arcs: Static-0 Hazards and Functional Hazards.

State transitions file (.gv)

The tool analyzes an AFSM design file containing the state **transition's logic expressions** and produces the Hazard Detection file.

Hazard Detection file (.txt)

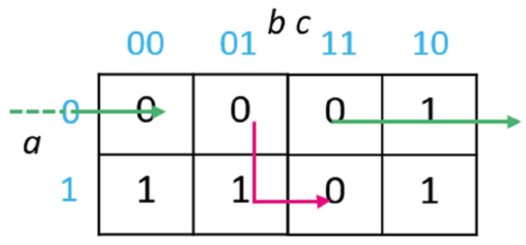
A list of all the **Static-0 and Functional Hazards** found in the Input net, and how to trigger them. From this file, the Hazard Testbench is created.

Hazard Testbench (.sv)

A **testbench** that specifically triggers the detected hazards. It is used to simulate them using commercial tools such as Cadence's xcelium.

3. MAIN STEPS

The Hazards Detection Tool was developed in Python. For each arc expression of the AFSM state transition file the tool:



- Generates the Karnaugh Map of the function
- Detects **Static-0 hazards**: the same input is repeated in the arc condition. Solved by rewriting the expression as sums of products.
- Detects **Functional hazards**: 2 variables changing correspond to 3-squares paths in the Karnaugh map. A 0-1-0 path is a possible glitch to 1.
- Prints a Hazard Detection file (.txt) with the Static-0 and Functional Hazards
- Generates a System Verilog Testbench stimulating each hazard found by the tool

4. RESULTS

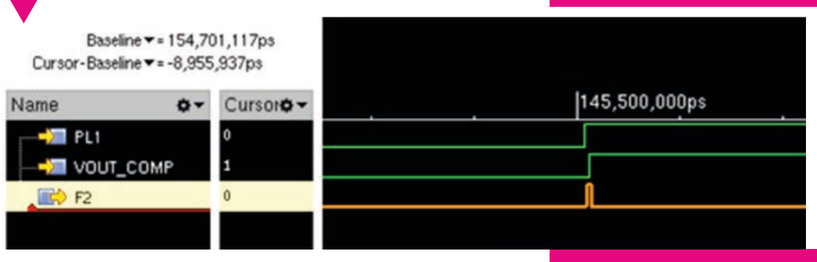
```
digraph AFSM_VO3 {
STATE1 -> STATE2 [ label =  " EN_VO3 " ];
STATE2 -> STATE3 [ label =  " PL1 & VOUT " ];
STATE2 -> STATE4 [ label =  " PL1 & ~ VOUT " ];
```

State transitions file (.gv)

```
FUNCTIONAL HAZARDS in AFSM_VO3
-----------------------------------------------
Arc : STATE2 -> STATE3: [ PL1 and VOUT]

VOUT: 0->1   and then    PL1: 1->0
PL1: 0->1    and then    VOUT: 1->0
```

Hazard Detection file (.txt)



Hazard Testbench (.sv)

CONCLUSIONS

Hazards inside the AFSM logic can trigger unwanted transitions, so they must be detected and solved.

The Hazard Detection Tool automatically detects all the dangerous hazards in the AFSM Arc Transitions Logic.

It prints the identified hazards in a .txt file and generates a .sv testbench stimulating each found hazard.

This method allows to focus on the simulation of critical cases. By triggering the hazards, it is possible to study their repercussion on the whole system.

The automation of the Hazard Detection process is a valuable part of the AFSM design. It significantly reduces the testing time and, consequently, the time-to-market and improves the quality of the product.