Adaptation of an Event-Driven Simulation Environment to Sequentially Propagated Concurrent Fault Simulation

Mina Zolfy, Shahrzad Mirkhani and Zainalabedin Navabi
Electrical and Computer Engineering / University of Tehran
{mzolfy, shahrzad} @cad.ece.ut.ac.ir; navabi@ece.neu.edu

Abstract

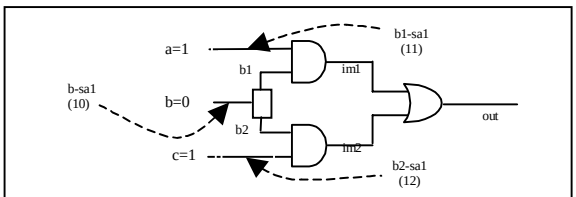
A new fault simulation method is presented here. The method relies on simulation cycle timing of event-driven simulators (delta delays in VHDL). This timing is used for propagation of faulty values in faulty sections of a circuit. This method is based on concurrent fault simulation and is implemented in VHDL. VHDL gate models that are capable of propagating faults in fault queues perform this fault simulation. Gate models process their fault queues and propagate them in delta time units. In these models, gates with faulty input values are expanded in delta time to evaluate faulty output values and propagate them to other sections of the circuit. Using ISCAS benchmarks, a performance improvement of up to 500X over serial fault simulation has been obtained. This work is useful for fault simulation of post-synthesis VHDL outputs.

Fault simulation methodology

The VHDL-based fault simulation method discussed here is referred to as Sequentially Propagated Concurrent Fault Simulation (SPC-FS). Implementing fault simulation in an event-driven simulation environment like VHDL requires circuit components or primitive gates to be sensitive to faults on their inputs as they are sensitive to changing like values in an ordinary simulation. A primitive gate that receives a fault on its input propagates the fault to its output unless it is masked by its good value or by faults on gate's other inputs (the latter situation occurs when gate is on a reconvergent point of a fanout). Each gates receives a list of faults on its inputs and processes each fault independently and sends a list of propagatable faults to its output. These lists are referred to as fault queues. A queue contains a *begin marker* (that also contains the line good value), fault elements, and an *end marker*. A fault element consists of a faulty value and fault's unique id. In addition to propagating faults at its inputs, a fault simulation gate model adds its own faults (the faults that are injected on gate's inputs) to the queue of faults on its output. This is implemented in VHDL for a gate model the behavior of which is summarized below:

1. Waits for good values and fault events on its inputs.
2. Stores received faults in linked lists.
3. Repeat 1 and 2 until *end markers* are received on all inputs.
4. Gate calculates good output value and sends it out.
5. For each fault in the generated linked lists: calculates faulty outputs according to faulty inputs, and sends it out if it was different from the good output.
6. Injects faults that belong to the gate's inputs and sends them out if different from the good output value.

Figure 1 shows a simple circuit and its corresponding faults propagated through circuit lines.



Delta time	a	b	c	b1	b2	imı	im2	out
0	x,-∞	x,-∞	x,-∞	x,-∞	x,-∞	x,-∞	x,-∞	x,-∞
1	1,0	0,0	1,0	…	…	…	…	…
2	1,-1	0,-1	1,-1	0,0	0,0	…	…	…
3	…	…	…	1,10	1,10	…	…	…
4	…	…	…	0,-1	0,-1	…	…	…
5	…	…	…	…	…	0,0	0,0	…
6	…	…	…	…	…	1,10	1,10	…
7	…	…	…	…	…	1,11	1,12	…
8	…	…	…	…	…	0,-1	0,-1	…
9	…	…	…	…	…	…	…	0,0
10	…	…	…	…	…	…	…	1,10
11	…	…	…	…	…	…	…	1,11
12	…	…	…	…	…	…	…	1,12
13	…	…	…	…	…	…	…	0,-1

Figure 1: Creation and propagation of fault queues

References:

[1] M. Abramovici, M. A. Breuer, A. D. Friedman, "Digital System Testing and Testable Design," Computer Science Press, 1990.

[2] Z. Navabi, "VHDL: Analysis and Modeling of Digital Systems," Mc Graw Hill, 1998.

[3] E. G. Ulrich and T. G. Baker, "Concurrent Simulation of Nearly Identical Digital Networks," Computer, Vol. 7, No. 4, pp. 39-44, April 1974.