ELEC 5280/6280 BIST

Assignment #4 Random Scan Test Patterns

Read "Circular BIST Tools for ASL Descriptions" on the "AUSIM software, manuals, and notes" link on the class web page (we will be using these tools more in the future but on the flip-flop extract tool, *cbistext.exe*, for this assignment).

Generate 200 random test patterns for scan design testing of your circuit using the random vector generator but this time you will use some additional options to generate scan vectors. Note that there are no scan flip-flops in your circuit but AUSIM can simulate your circuit as if scan design were implemented (but without the shifting in of the test vectors so the simulation should be fairly fast). Use the following parameter and options:

ranvec *s#.vec* 200 *#ins* scan *#ffs*

where s# is the name of your assigned circuit, *#ins* is the number of primary inputs to your circuit (not counting the clock input), and *#ffs* is the number of DFFs in your circuit. You can get the *#ffs* number by first generating the scan chain file for your circuit by typing:

cbistext *s#.asl s#.scn*

The *s#.scn* file generated by this command will be needed by AUSIM for your scan chain flip-flop ordering during the simulation. For this assignment, we will use the default ordering produced by the *cbistext* program. Note that the *ranvec.exe* program will generate 201 files (*s#.vec* plus 200 scan input vector files, *scan0-scan199*). During logic simulation a scan output vector file will be generated for each scan input vector file, labeled *scan0o-scan199o*, which give the output results from the scan out operation. Note that after completing all of your simulations for this assignment you will want to delete the scan# files since there will be 400 of them. They can easily be reproduced by executing *ranvec.exe* followed by a logic simulation. Make sure your vector set *s#.vec* prefix (s#) matches your ASL name *s#.asl*.

A. Record the number of flip-flops in your circuit.

B. Run a logic simulation. Did all primary outputs initialize (look at logic simulation results in s#.out)?

C. Run a parallel fault simulation for both collapsed and uncollapsed single stuck-at gate-level faults and run a serial fault simulation for collapsed dominant bridging faults. Record the following data for each of the three fault simulations:

1. Fault simulation time:
2. Total number of faults:
3. Number of faults detected:
4. Number of potentially detected faults (and number of oscillation faults in the case of bridging faults):
5. Fault coverage (assuming 0.5 probability of detection for potentially detected and oscillation faults):

D. Calculate the area overhead for implementation of full scan in your circuit in terms of:

1. #G - assume 3 gates/MUX and 9 gates/FF (note that DFFs are only counted as 1 gate in the AUSIM audit)
2. #G_{IO} - assume 9 G_{IO}/MUX and 25 G_{IO}/FF (note that DFFs are only counted as 4 G_{IO} in the AUSIM audit)

Be sure to show your calculation work (indicate the method your are using to calculate area overhead).

Turn in your results on paper at the beginning of class on or before the specified deadline.

Happy BISTing!!