

ELEC 5280/6280 BIST

Assignment #6 Implementation of a Built-In Self-Test for Manufacturing Testing

In this assignment, **DO NOT** use a single XOR gate with more than two inputs; instead construct a linear or balance tree of 2-input XOR gates. You *may* use the functional XOR gate model provided in AUSIM, however, you should keep in mind that the fault coverage is not accurate since the pin fault model for an XOR is optimistic in that only 3 of the 4 possible vectors are needed to detect pin faults while all four vectors are needed to detect the faults associated with any gate level implementation of an XOR. As a result, using a single XOR gate with greater than 2 inputs gives overly optimistic fault coverage results. For you assigned circuit:

1. Design a maximal length TPG (LFSR or CAR) with number of flip-flops, $N_{FF} = \max\{11, N_{IN}\}$, where N_{IN} = the number of primary inputs to your circuit (not counting the clock input). The TPG should have an active low synchronous preset which will be connected to an active high BIST signal. Write an ASL description for the TPG with input ordering: CLK, PRESET. Verify the operation of your TPG in AUSIM (and debug as needed).
2. Design a 16-bit SAR with primitive polynomial and active low clear which will also be connected to the active high BIST signal. Write an ASL description for the ORA with input ordering: CLK, CLEAR, DATAIN. Verify the operation of your ORA in AUSIM (and debug as needed).
3. If the number of primary outputs of your circuit is more than one, design a N -bit concentrator, where N = number of primary outputs of your circuit. Write and ALS description for you concentrator and verify the design in AUSIM.
4. Write a top-level ASL description that incorporates your TPG, ORA, and concentrator (is applicable) with your assigned circuit with input ordering (CLK, BIST) and output ordering (SAR0-15). Note you will need to include the ASL description of your four (or three circuits if no concentrator is needed) and convert them from *ckt:* to *subckt:*. Also note that the primary I/O of your original circuit are now internal nodes that can only be controlled and observed by the TPG and ORA, respectively. *If your circuit has a global RESET signal, you may connect this signal to your BIST input to use to initialize the CUT while connecting the remaining inputs to the original circuit to the TPG.*
5. Generate a set of test vectors to initialize and initiate the BIST sequence and then to run the BIST sequence for 2000 clock cycles. Perform a logic simulation to determine whether your circuit will initialize and stay initialized. If the circuit will not initialize or stay initialized, try keeping the BIST signal inactive for more clock cycles (this will keep the SAR cleared while the TPG has a chance to initialize the circuit). If this does not work, generate a scan (.scn) file and add a pseudo-scan vector in the vector set while keeping the BIST signal inactive. Recall that a scan file can be generated by “cbistext *bist.asl bist.scn*” you will have a scan file for your flattened circuit with TPG and SAR (include ‘keepfas’ option after ‘proc’ in AUSIM control file to save a flattened ASL file for use with cbistext, but you must use cbistext v1.1 included in the new AUSIM.zip). The beginning of the vector set can be produced by “ranvec *bist.vec 1 2 scan #ffs*”. Note that you will need to keep BIST inactive for at least one clock cycle after the scan vector to preset the TPG and clear the SAR. Add your 2000 clock cycles after the initialization sequence.
6. Run a parallel fault simulation for the complete circuit using collapsed gate-level stuck-at faults. The complete circuit includes subcircuits for s#, your TPG design, your ORA design, and your concentrator design (if you needed a concentrator). Look at the fault list and you will see the faults are partitioned into four (or three if no concentrator was needed) groups with the instantiation name as a prefix for each gate name in the fault list. As a result, the detect, undetected, and potentially detected fault lists will preserve this naming convention and can easily be parsed to record the following for the complete circuit simulation:
 - For the complete circuit and for each of the four (or three) subcircuits record/calculate:
 - A. Total number of faults:
 - B. Number of faults detected:
 - C. Number of potentially detected faults:
 - D. Fault coverage assuming probability of potential detection = 1:
 - Calculate the area overhead (by dividing the total with BIST as reported by AUSIM by the total in the original circuit w/o BIST as reported by AUSIM) in terms of the following:
 - A. #G - assume 9 gates/FF (note that DFFs are only counted as 1 gate in the AUSIM audit)
 - B. #GIO - assume 26 GIO/FF (note that DFFs are only counted as 4 GIO in the AUSIM audit)

Email your complete hierarchical ASL and input vector set to me before the beginning of class and turn in your results on paper at the beginning of class on or before the specified deadline. As always, Happy BISTing!!