Nathaniel Jansen (njansen)

Elliot Rosen (elliotr)

Project 5

For our DFT we implemented boundary scan, internal scan, and BIST. Boundary scan was run of the mill. For internal scan, we used a full internal scan, allowing us to completely control state. Similarly, for BIST, we have a 247 bit LFSR that is connected to all of the PIs and internal scan FFs allowing us to generate pseudo random vectors that control the entire state and all of the inputs. It has two XOR networks to compress the BIST output data. One is connected to the internal scan registers, the other is connected to the POs. These XORs are then XORed together and the result is visible on TDO

We did not successfully generate test patterns, however we theoretically can get 99.9% test coverage. We generated test vectors for the combinational version of the circuit and we can apply these through a combination of internal scan and boundary scan. Additionally, BIST can mop up what we missed.

For BIST, we would need the following sequence of test patterns:

T = 0: TRST = 0, TMS = 1, TDI = 0; (PI=X)

T=10 (make sure we go to the right state): TMS = 0

T=11: TMS =1

T=12: TMS=1

T=13: TMS = 0

T=14:TMS=0

T=15:TMS=0, TDI = 1

T=16:TMS=1, TDI=1

T=17:TMS=1, TDI=0

18: TMS=0, TDI=0

19:TMS=0, TDI=0

20:TMS=0, TDI=0

Hold these values steady until BIST is done (~30,000 cycles is what we decided on)

When you’re ready to end BIST, do the following sequence:

TMS=110001111…. (where the 1’s hold it in the normal state

TDI=0000000000….

The instruction op codes are:

00 = bypass register

01 = boundary scan

10 = internal scan

11 = BIST

COST:

|  |  |
| --- | --- |
| **Number of BIST cycles (B)** | **30,000** |
| **Number of non-BIST cycles (S) and tester channels (TS)** | **(28,000, 5)** |
| **Total number of input and outputs added (P)** | **4** |
| **Number of tester channels (T)** | **5** |
| **Total number of normal (DN) and reset (DR) flip-flops added** | **(510, 11)** |
| **Total number of gate inputs added to s9234 (G)** | **6000** |
| **Total Cost (C)** | **41562** |

The non BIST cycles number is probably an overestimate, as BIST will probably get the coverage of a lot of those vectors, meaning we’re probably actually under the cost threshold.

Files in sourcefiles directory:

BIST.v: lfsr implementation

DFT.v: implementation of modules used in TAP architecture diagram from lecture notes

Primitives.v: provided file, with some added primitives

S9234.v: File with all DFT implemented

S9234\_comb.v: Combinational version of s9234 for use in TP generation

S9234\_orig.v: original circuit (used for “normal mode works” TB)

TAP.v: tap controller

boundaryTB: testbench that tests boundary scan cell functionality

lfsrtb.v: testbench for lfsrtb

normalOpTB.v: tests normal mode

s9234\_bistTB: successfully puts circuit in BIST mode, runs for a while, stops

TAP\_ir\_tb: tb to test sending instruction to IR through tap controller.

Directory in sourcefiles directory:

“attempt at TPG and Encounter” – some evidence of our attempt at using encounter.