

Circular BIST Tools for ASL Descriptions

These tools work with the ISCAS'89 sequential logic benchmark circuits in ASL (Auburn Simulation Language) form and can be obtained in the AUSIM.zip file on the AUSIM webpage. The library file "cbist.lib" must be included as the lib file in the AUSIM control file; this library provides the circuit descriptions for the CBIST components inserted by the test synthesis software. Software for implementing CBIST with input isolation MUXs is typically used in the following order:

```
cbistext <in_asl> <ff_file>
```

This program extracts the flip-flops from the ASL circuit description for manipulation and insertion of CBIST. The "in_asl" parameter refers to the original circuit description and the "ff_file" gives the name of the file the list of flip-flops are to be written into. Once the flip-flops have been extracted the list can be modified for re-ordering or selective replacement. A random ordering can be obtained for the flip-flop list with:

```
cbistran <ff_file> <decimal_seed_number>
```

This program takes the original flip-flop file and randomly orders the list in the file "ff_file" using the "decimal_seed_number" to initialize the pseudo-random number generator. The reordered list of flipflops is written back into the file "ff_file".

Next the CBIST implementation is inserted in to the circuit description with:

```
cbistnr <in_asl> <out_asl> <ff_file> <num_muxff>
```

This program implements CBIST with input isolation MUXs and/or SRs. It also inserts CBIST FFs and orders the chain according to the list of FFs given in "ff_file". The original circuit description is maintained in "in_asl" and the circuit description with the CIBST implementation is written into "out_asl". The number of input isolation SR bits inserted into the circuit description is specified by the "num_muxff" parameter where any additional primary inputs (beyond the number specified) will have isolation MUXs inserted.

The following program produces a set of input vectors for CBIST logic and fault simulations which includes the initialization sequence:

```
cbistvec <vec_file> <num_ins> <num_vecs> <num_ffs> <num_reseed>
```

The "num_ins" parameter refers to the number of primary inputs excluding the clock input and CBIST control inputs, while the "num_vecs" parameter refers to the number of clock cycles desired in the CBIST sequence excluding the fixed initialization sequence. An associated *scan* file is generated to scan out the final CBIST signature at the end of the BIST sequence. The "num_ffs" parameter is the number of flip-flops in your circuit. The "#reseeds" indicates the number of times the CBIST chain is to be reseeded and additional *scan#* files are generated to reseed the CIBST chain with random state vectors periodically during the CBIST vector set.

The following program removes faults in the fault list that are associated with the CBIST circuitry.

```
cbistrmf <inflt_file> <ff_file> <outflt_file>
```

where "inflt_file" is the fault list generated for the ASL complete circuit description with CBIST implementation, "ff_file" is the flip-flop file obtained from *cbistext*, and "outflt_file" is the new fault file with CBIST circuitry related faults removed.

Happy CBISTing!

Charles Stroud, Professor
Dept. of Electrical & Computer Engineering
Auburn University
October 12, 2004