Logic Synthesis & Verification, Fall 2012

National Taiwan University

Programming Assignment 2

Due on 2012/12/12 before lecture

1 [Programming ABC]

Programming task:

Please write a procedure in the ABC environment to print the don't care statistics of a given `strash`ed circuit with the following instructions.

1. Name the command `print_sdc`.
2. Write your program in `abc.c` starting with a new function named `Abc_CommandPrintSDC`.
3. Use SAT-based computation to determine, for each (AIG) node, which of the input patterns $\{00, 01, 10, 11\}$ can never occur. Please take into account the polarity of each AND gate. For example, consider an AIG node c with two inputs from nodes a and b. Suppose the second input of c is inverted, and nodes a and b can never be 1 at the same time. Then pattern 10 is a don't care pattern for the inputs of node c.

Programming help:
One simple implementation may proceed in the following steps: 1) `strash` the current network into an AIG, 2) convert the AIG into CNF[1], and 3) enumerate through every node of the AIG and test its four input combinations by adding their corresponding **unit assumptions**[2] upon SAT solving[3].

Items to turn in:

(1) Your new file `abc.c`
(2) A brief description about your implementation, where any effort to enhance computation efficiency should be highlighted. (A 10% bonus credit will be given to efficient implementations.)
(3) A screenshot of ABC running your new command "`print_sdc`" on example `s444.blif`.

---

[1] Procedure `Cnf_DeriveSimple` is recommended for circuit to CNF conversion (easier to understand).
[2] Unit assumptions can be added in the second and third arguments of `sat_solver_solve` for incremental SAT solving.
[3] Procedure `Abc_NtkDSat` in `src/base/abci/abcDar.c` shows an example of SAT solver usage.