**Project: STATISTICAL STATIC TIMING ANALYZER**

# 1 Addendum to Statistical STA (SSTA) Project

The outline of the project have been described in lecture notes. This document covers some details to aid your implementation. The following can methods can be followed in your implementation:

- Read an easily accessible and easy to handle benchmark file format capable of moderate complexity. The ISCAS'89 benchmarks (`*.bench` file format), which are on the WebCT directory, are suggested. The file format is very simple and self-explanatory.
- Three sets of benchmark circuits are provided: (1) Combinational circuits with one primary output (PO), (2) Combinational circuits with multiple PO's, (3) Sequential circuits with multiple PO's. The expected complexity of SSTA tool implementation to handle these circuits are in ascending order.
- Assume the arrival times of the data signals at primary inputs (PI's) are zero.
- Assume two shared (global) sources of variation $\Delta x_1$ and $\Delta x_2$ and one local (independent) source of variation $\Delta x_R$ for all circuits. These variations are modeled with normal distributions $\mathcal{N}(0, 1)$.
- For simplicity, assume all registers in the (sequential) benchmark circuits are flip-flops (*i.e.* assume latches are flip-flops).
- Unless your chosen benchmark format comes with timing information, you must develop a reasonable timing model for your circuits. In the case of the `*.bench` files, there is *no* timing information provided with the benchmark circuits. Thus, arbitrary timing information are generated for these circuits; for gates and interconnects. As described in lecture notes, simplistic timing models are used for gates. These delay models for gates are provided in the file "*cell_library.time*". The timing information for the interconnects of a circuit "<*circuitname*>*.bench*" is provided in "<*circuitname*>*.time*". You can develop any model as long as it produces reasonable numbers, and contribute to the cell library.
- The effects of fanout in cell delay are ignored for simplicity, e.g. a two-input AND and a ten-input AND gate has the same delay. In cell_library.time, AND1, AND2 and AND3 do not indicate one, two and three input AND gates. They list three different AND implementations that can be used for any number of inputs. It certainly is an unrealistic assumption, however, it is one of the simplifications to make the project more feasible to be completed.
- Note the format of interconnect delay files:

  $$StartWire \quad StopWire \quad a_0 \quad a_1 \quad a_2 \quad a_R$$

  $StartWire$ is the interconnect on which the delay is measured. Note that an interconnect can fanout to different gates, which would (potentially) lead to different interconnect delays between the source gate and each one of the multiple sink gates. The $StopWire$ is given to identify the delay between the source gate and the target gate—$StopWire$ is the wire on the output terminal of the target gate.
- Next, your tool can be designed to calculate a reasonable number for the clock period.[1] The real hardship would be in computing the clock period for sequential circuits. In static timing analysis, this can be done by computing the longest paths from a register output to a register input throughout the circuit. For this purpose, Dijkstra's or Bellman-Ford's (BF) algorithms can be used (BF is better because it runs faster, it must be run on acyclic graphs, which yours will be because there are no combinational loops), and the longest path delay in the circuit is selected as the clock period. Think about (or research) how this analysis can be performed for SSTA. Implementation of such analysis will significantly improve the quality of your SSTA tool.

[1] Other alternative design objectives are presented on slides 30–32.