

Automatic Generation of Configuration Files for AccuCell

AccuCell provides an auto configuration command, `gen_cfg` that allows user to automatically create configuration files for a library and for each cell. A pre-existing Liberty format library is required to run this command and Figure 1 shows its function.

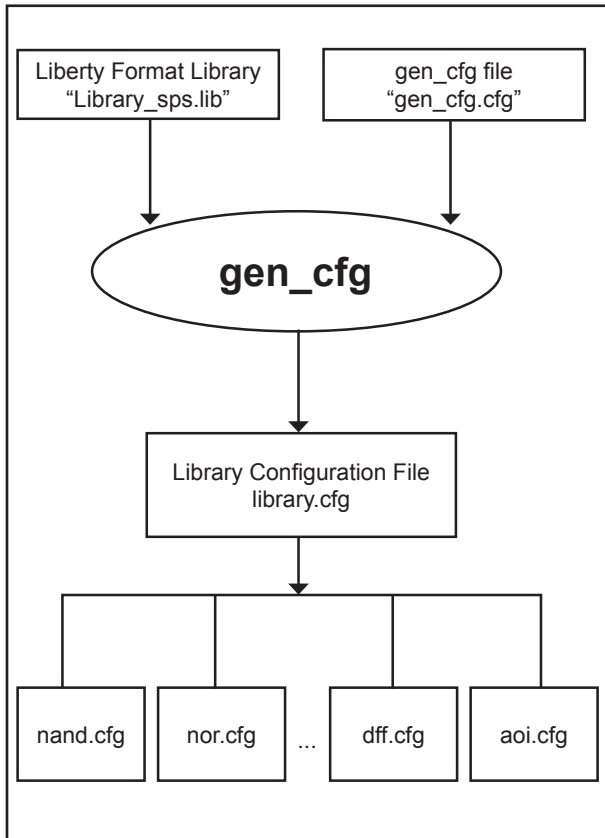


Figure 1. Inputs and outputs.

The command needs an argument as shown below:

gen_cfg *gen_cfg_file*

gen_cfg_file is required to run this command. Examples of this file is provided below. This file is generally used to control the contents of generated configuration files. • The POWERS, GROUNDS each define the global node for the PG rails of the cells.

```

POWERS VDD
GROUNDS VSS
IN_FILE_NAME_REL ./netlists
IN_SNPS_LIB_NAME ./library_sps.lib
SUBCKT_EXTN net
SLOPE 0
    
```

Figure 2. Example of "gen_cfg_file".

- The POWERS and GROUNDS each define the global node for the PG rails of the cells.
- IN_FILE_NAME_REL define relative path to an individual file containing ALL cell netlists as individual cell .subckt definitions, OR it may be a relative path to a directory containing a set of netlists for each cell. In the latter case, SUBCKT_EXTN should also be used.
- IN_SNPS_LIB_NAME define a path to Liberty library "Liberty_sps.lib"
- SUBCKT_EXTN can be used to define the <ext> extension of the netlist for each cell in the form of <cell_name>.<ext>.
- SLOPE 0 turns off the defining of SLOPE_TABLE slope values in the cell .cfg file so as to ignore the pre-defined slopes in the .lib the read by gen_cfg. Setting SLOPE 1 (or leaving it out of the gen_cfg_file) includes slope definitions in each cell as defined in the .lib file read by gen_cfg.

To run `gen_cfg` and generate configuration files, create a `gen_cfg.tcl` file as follows:

```
gen_cfg gen_cfg.cfg
```

Lastly, at the system prompt type:

```
accucell gen_cfg.tcl |& tee gen_cfg.log
```

The following example illustrates the result of running `gen_cfg` for a library containing cells INVX1 and INVX2.

```
[kaoruk@sunrayjp test]$ ls
INVX1  INVX2  gen_cfg.cfg  gen_cfg.tcl  library_sps.lib  netlists
[kaoruk@sunrayjp test]$ accucell  gen_cfg.tcl | & tee gen_cfg.log
Loading ByteCode Loader...
AccuCell(R) 2.6.18.C. Mon Nov  3 12:56:53 PST 2014, Proprietary and Confidential Software
Copyright (c) 1984 - 2015. Silvaco Inc.

powers vdd
grounds vss
in_file_name_rel netlists
in_snps_lib_name  ./library_sps.lib
snps_lib_path: ./library_sps.lib
subckt_extn .net
./library_sps.lib read successfully
Processing library Cell_EX2 with 2 cells
[kaoruk@sunrayjp test]$ ls
Cell_EX2.cell_list  INVX1  delay.1  gen_cfg.cfg  gen_cfg.tcl      netlists
Cell_EX2.cfg        INVX2  delay.2  gen_cfg.log  library_sps.lib
```

Figure 3. Generated files from auto-generation command “gen_cfg”.

```
...
...
cell (inv) {
  area : 0;
  pin (A) {
    direction : input ;
    capacitance : 0.01167;
    rise_capacitance : 0.01166;
    fall_capacitance : 0.01167;
    rise_capacitance_range (0.01166 , 0.01166) ;
    fall_capacitance_range (0.01167 , 0.01167) ;
    clock : false;
    max_transition : 1.0;
  }
  pin (Y) {
    direction : output;
    max_capacitance : 0.29498;
    function : “!A”;
    internal_power () {
      related_pin : “A”;
      rise_power (pwr_template4x4) {
        index_1 (“0.12500, 0.25000, 0.50000, 1.00000”);
        index_2 (“0.01000, 0.02000, 0.04000, 0.08000”);
        values (“0.04925, 0.06377, 0.09399, 0.15582”, “0.06145, 0.07462, 0.10282, 0.16216”, \
              “0.08851, 0.10006, 0.12573, 0.18125”, “0.14490, 0.15451, 0.17700, 0.22763”);
      }
      fall_power (pwr_template4x4) {
        index_1 (“0.12500, 0.25000, 0.50000, 1.00000”);
        index_2 (“0.01000, 0.02000, 0.04000, 0.08000”);
        values (“0.02038, 0.03705, 0.06922, 0.13245”, “0.00949, 0.02813, 0.06254, 0.12779”, \
              “0.01637, 0.00482, 0.04329, 0.11335”, “0.07218, 0.04830, 0.00485, 0.07340”);
      }
    }
  }
  timing () {
    related_pin : “A”;
    timing_sense : negative_unate;
    cell_rise (delay_template4x4) {
      index_1 (“0.12500, 0.25000, 0.50000, 1.00000”);
      index_2 (“0.01000, 0.02000, 0.04000, 0.08000”);
      values (“0.07676, 0.10078, 0.15069, 0.25288”, “0.11439, 0.14147, 0.18932, 0.28745”, \
            “0.18412, 0.21727, 0.27138, 0.36649”, “0.31137, 0.35516, 0.42304, 0.53144”);
    }
  }
}
```

Figure 4. Liberty Library Sample “sample_library_sps.lib”.

Conclusion

Depending on the nature of the Liberty.lib file used, the generated configuration files may require modification prior to running characterization. However, in most cas-

es all required configuration setup will be complete with ready-to-run files. Therefore a significant productivity boost for most characterization efforts will result.