Usage : Iterative Failure Region Diagnosis Program (IFRD)

DAVID HOSOON SHIN

I. Introduction

'IFRD' is the stochastic failure analysis program with iterative manner, which is implemented by python language. This program is designed to analyze the extreme failure region and provide the guideline to meet the target of design successfully.

II. Major features

"IFRD" includes those major features

- Data sampling based on
    1. SOBOL sequence
    2. Gaussian distribution
    3. Uniform distribution
- Data learning and classification using
    1. Gaussian Radial basis Function
    2. KNN distance
    3. Neural network
- Automated generation of SPICE deck and simulation using NGSPICE
    - Sample circuits listing
        1) 6T-SRAM circuit
        2) Charge pump & filter
        3) 4-stage logic gate
        4) CMOS inverter
- Curve fitting with simulated data
    1. Gaussian distribution
    2. Log normal distribution
    3. Pareto distribution
- Elite sampling to filter out majority data considering
    1. Correlation coefficient of data
    2. MIN/MAX distance from median value on the result distribution.
- Feature selection methods to do "in-spec" validation using several methods
    1. Correlation coefficient based
    2. Correlation coefficient after pruning data based on "Relief" method
    3. Variance based
    4. Tree-based feature selection (using ensemble learning)

III. Code description and location

'IFRD' testing environment was built in two major parts. Main program invokes running NGSPICE inside to simulate sample circuit and getting result

1) Main program location: /home/locker/EE/mscad/cvs_repos/david/workspace

2) Simulation running and result stored location:
/home/locker/EE/mscad/cvs_repos/david/REscope/ChargePump | CP (Inverter) | LOGIC | SRAM /SIM

Corresponding to test circuits, the location of program installed is as follows:

1. Charge Pump
   - ➢ Main program location :

     /home/locker/EE/mscad/cvs_repos/david/workspace/DAC_ChargePump/src

       - IFRD.py : Main program to run
       - REclass.py : All classes implemented inside
   - ➢ Simulation location :

     /home/locker/EE/mscad/cvs_repos/david/REscope/DAC_ChargePump/SIM

       - cp_run.sp : SPICE input deck for Charge Pump (called by main program)
       - mos.model : BSIM4 model for MOS process variation (included by cp_run.sp)

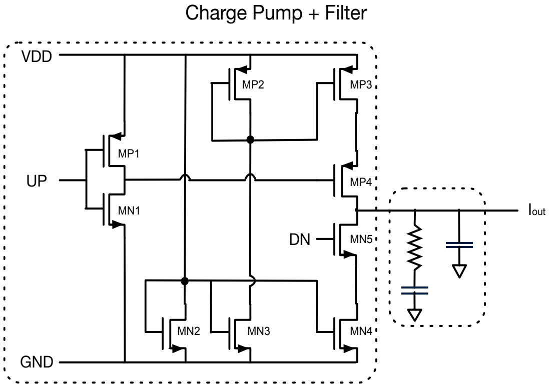
Charge Pump + Filter

Fig. Charge pump & filter architecture

2. 6T-SRAM
   - ➢ Main program location :

     /home/locker/EE/mscad/cvs_repos/david/workspace/DAC_SRAM/src

       - IFRD.py : Main program to run
       - REclass.py : All classes implemented inside
   - ➢ Simulation location :

     /home/locker/EE/mscad/cvs_repos/david/Rescope/SRAM/SIM

       - 6t_sram_run.sp : SPICE input deck for 6T-SRAM (called by main program)
       - mos.model : BSIM4 model for MOS process variation (included by 6t_sram_run.sp)

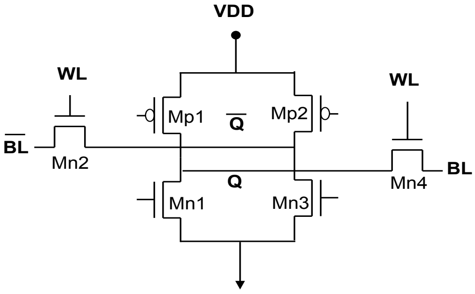

Fig. 6T-SRAM architecture

3. 4-stage logic path
   - Main program location :

   /home/locker/EE/mscad/cvs_repos/david/workspace/LOGIC_REstat_ELSS/src

     - REstat_ELSS.py : Main program to run
     - REclass.py : All classes implemented inside
   - Simulation location :

   /home/locker/EE/mscad/cvs_repos/david/Rescope/LOGIC/SIM

     - logic_run.sp : SPICE input deck for 4-stage logic path (called by main program)
     - mos.model : BSIM4 model for MOS process variation (included by logic_run.sp

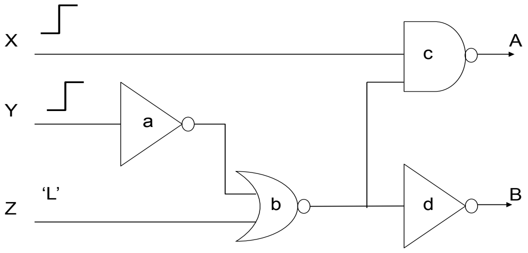

Fig. 4-stage logic path architecture

4. CMOS Inverter
   - Main program location : /home/locker/EE/mscad/cvs_repos/david/workspace/CP_REstat_ELSS/src
     - IFRD.py : Main program to run
     - REclass.py : All classes implemented inside
   - Simulation location :

   /home/locker/EE/mscad/ cvs_repos/david/Rescope/CP/SIM

     - cp_run.sp : SPICE input deck for a CMOS inverter (called by main program)
     - mos.model : BSIM4 model for MOS process variation (included by cp_run.sp

- Common file for SOBOL sequence data generation : sobol_seq.py
- Every details about class and function has been commented inside files (*.py)

IV. Running Guide

1. Python Package install
   - ➢ 'IFRD' uses python interpreter and library packages as follows:
     - python 2.7: Python interpreter version 2.7
     - numpy: Numerical python package for all data type and manipulation, linear algebra
     - scipy : Scientific python package for statistical processing
     - sklearn : Library for data learning and classification
     - pybrain : Library for Neural network building.
     - matplotlib: Data plotting package like 'matlab' style

   - ➢ All information about above packages are provided in following links
     - python2.7: http://www.python.org/download/releases/2.7
     - numpy: http://www.scipy.org/scipylib/download.html
     - scipy : http://www.scipy.org/scipylib/download.html
     - sklearn : http://scikit-learn.org/stable/
     - pybrain : http://pybrain.org
     - matplotlib: http://matplotlib.org

2. NGspice install
   - ➢ 'IFRD' uses NGspice for the circuit simulation.
   - ➢ Official site for NGspice : http://ngspice.sourceforge.net

- All environmental setup for python packages and NGspice has been setup in Newton server.

3. Running Command (In terminal)
   - ➢ Running in same place where all source files (*.py) are located.

```
Usage: python Main_program_name.py (e.g. IFRD.py, REstat_ELSS.py)
```