SLiCAP definition and execution of an instruction

```
from SLiCAP import *
# Create an instance of the project class.
"""
This initializes the project paths, creates the directory
structure, a main index HTML page and compiles the
system libraries."""
my_prj = SLiCAP.initProject( < projectName >)
# Create an instance of the instruction class
my_instr = SLiCAPinstruction.instruction
# Check a netlist and define the circuit from it for the instruction
SLiCAPinstruction.instruction.setCircuit( < fileName > )
# Define the simulation type, the gain type and the data type
my_instr.setSimType( < simulation_type > )
my_instr.setGainType( < gain_type > )
my_instr.setDataType( < data_type > )
# If the simulation type is set to 'numeric' define the parameters
# You would like to keep symbolic in the results.
my_instr.keepSymbolic( < list with parameter names > )
# Execute the instruction.
""" The results will be stored in a result dictionary with key-value pairs:
key:
name of the result, automatically generated by SLiCAP
value:
expression."""
# How to deal with pole-zero analysis?
# Automatic generation of names?
# Results dict and units dict?
```