

How to run schematic level simulation in Keysight ADS using Python

Ver. ADS 2023

Hyeon-Yeong Yeo

1. Extract netlist from schematic.

In the schematic window,

```
DynamicLink > Top-Level Design Netlist
```

2. Find the location of your ADS and set *HPEESOF_DIR* before simulating

Option 1. Bash file before running the script

Example) in .bashrc file

```
export HPEESOF_DIR=/usr/local/packages/ADS2023
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:
/home/local/ace/hy7557/miniconda3/envs/myenv/lib:$HPEESOF_DIR/lib/
linux_x86_64:$LD_LIBRARY_PATH
```

→ run the bash script before you run simulation.

```
source .bashrc
```

Option 2. (Python Script ver.) Include in subprocess commands inside your python script.

Refer to *setup_commands* in the script below.

```
def run_simulation(modified_netlistfile_dir, tb_name, wrk_space):

    # Define the base setup commands
    setup_commands = [
        'cd '+ wrk_space
        'set "HPEESOF_DIR=/usr/local/packages/ADS2023"',
    ]

    # Combine the base setup commands into a single command string
    setup_command_str = ' && '.join(setup_commands)

    my_env = os.environ.copy()
```

```

fullCommand = f"{setup_command_str} &&
\"/usr/local/packages/ADS2023/bin/adssim\" {modified_netlistfile_dir}
"

process = subprocess.Popen(fullCommand, shell=True,
stdout=subprocess.PIPE, stderr=subprocess.PIPE, env=my_env)
stdout, stderr = process.communicate()

# # Print the output and error (if any)
# print(f'Output for {netlist}:\n{stdout.decode()}')
# if stderr:
#     print(f'Error for {netlist}:\n{stderr.decode()}')

# Decode the output
stdout = stdout.decode(errors='replace')
stderr = stderr.decode(errors='replace')

# Check for errors in the simulation process
if "hpeesofsim terminated due to an error" in stderr or "hpeesofsim
terminated due to an error" in stdout:
    raise RuntimeError(
        f"[Simulation Error] Simulation terminated for
{tb_name}.\n"
        f"STDOUT:\n{stdout}\n"
        f"STDERR:\n{stderr}"
    )

dsfile = os.path.join(wrk_space, tb_name+".ds")

return dsfile

```

3. Find the location of the ADS simulation (.exe) file.

Example) `\"/usr/local/packages/ADS2023/bin/adssim\"`

*In Windows, the simulation file is `adssim_win.exe`

4. Run simulation with your netlist.

For example, if your netlist file is “*netlist.log*” (default) and your ads simulation exe file is `usr/local/packages/ADS2023/bin/adssim`

Option 1. In command line

```
usr/local/packages/ADS2023/bin/adssim netlist.log
```

Option 2. In Python script

```
dsfile =  
run_simulation(modified_netlistfile_dir=modified_netlist,  
tb_name=tb_name, wrk_space=wrk_space)
```

*Regardless of the name of the netlist file, the output (ds file) will have a name following the testbench name. For example, if your netlist file *netlist.log* starts with the line:

```
; Top Design: "rf_rx_lib:TB_Manual3:schematic",
```

the output (ds file) will be made in the name of “*TB_Manual3.ds*”.

How to read dsfile (ASCII file) using Python

1. Find the location of python provided by Keysight.

Example) `/usr/local/packages/ADS2023/tools/python/bin/python3.10`

2. Make a python script using python library from ADS.

* key functions:

1) `dataset.open(dsfile)`

2) `output_data[member].to_dataframe().reset_index()`

```
import keysight.ads.dataset as dataset  
from keysight import pwdatatools as pwdt  
  
output_data = dataset.open(dsfile)  
for member in list(output_data.keys()):  
    block_data = output_data[member].to_dataframe().reset_index()  
    block_cols = block_data.columns
```

In this way, you can access the data in the dsfile in the form of pandas dataframe.

Example)

```
dsfile = final_stage_dsfile  
output_data = dataset.open(dsfile)  
# output_into_group = pwdt.read_file(dsfile)
```

```

for member in list(output_data.keys()):
    block_data = output_data[member].to_dataframe().reset_index()
    block_cols = block_data.columns
    # print(f"block_data:{block_data}", file=sys.stderr)
    # print(f"block_cols: {block_cols}", file=sys.stderr)

    if ("case1_Vin_fund" in block_cols): # ALL VIN_FUND data are in
the same block
        freq_vin_fund = (block_data["RFfreq"]*1e9).round(2) # Hz
        case1_vin_fund = block_data["case1_Vin_fund"] # V
        case2_vin_fund = block_data["case2_Vin_fund"] # V
        case3_vin_fund = block_data["case3_Vin_fund"] # V

```

3. (Optional) If you want to run the script with your original python kernel without switching to python ADS, you can follow the script below:

```

ads_python = "/usr/local/packages/ADS2023/tools/python/bin/python3.10"

cmd = [ads_python,
"/home/local/ace/hy7557/rf_rx_darts/read_ads_dsfile.py", dsfile]

try:
    result = subprocess.run(cmd, capture_output=True, text=True,
check=True)
except subprocess.CalledProcessError as e:
    print("[ERROR] Simulation script crashed.")
    print("Command:", " ".join(cmd))
    print("Return code:", e.returncode)
    print("STDERR:\n", e.stderr)
    print("STDOUT:\n", e.stdout)
    raise RuntimeError("Simulation script failed.") from e

```