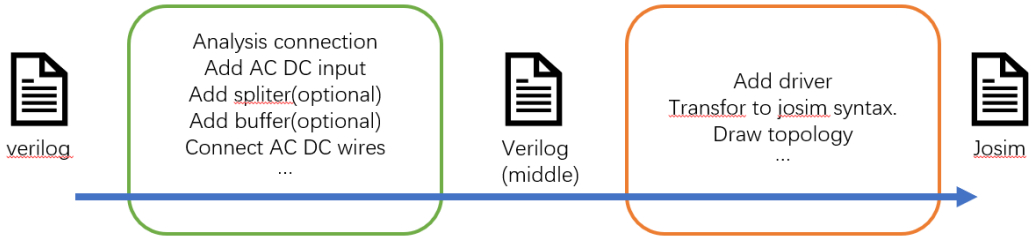
V2V2J Manual (Verilog to Verilog to JoSIM):

Description:

The framework of V2V2J is:



Firstly, you need to install:

Windows program (or other OS):

graphviz, https://graphviz.org/download/

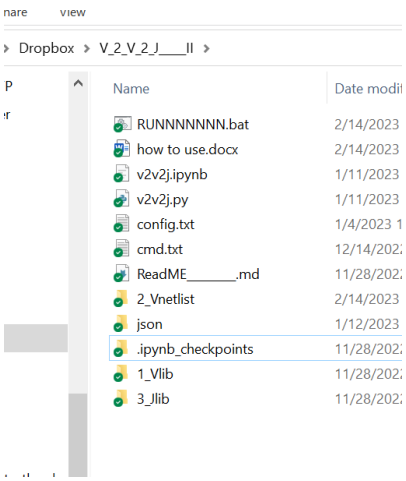
Python library:

graphviz (python-interface): python -m pip install graphviz

networkx, pip install networkx[default]

tqdm, pip install tqdm

This is the directory of all files.



The **v2v2j.py** is the core program.

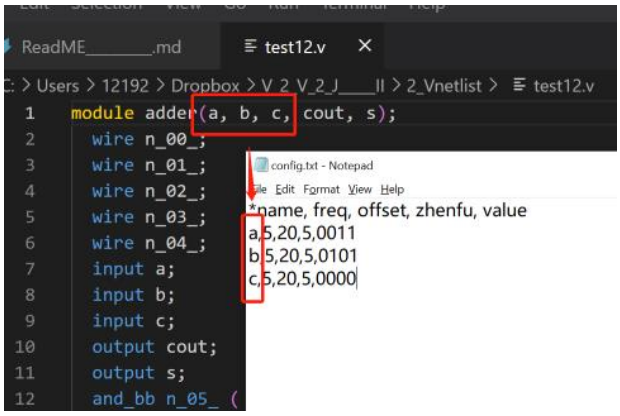
The folder:

1_Vlib: place the Verilog library. Usually no need to change.

2_Vnetlist: place the Verilog file you want to transfer into josim sim file.

3_Jlib: place the Josim library. Usually no need to change.
Json: initial is empty, it will be placed some parsed files while running v2v2j.py. (a json file, store all cells info extract from verilog).

The files:
config.txt: is the config of writing drivers (AC, DC voltage, offset…).
You should set the parameters for your purpose. But make sure the "name" should same with your Verilog input. Like this figure:



Before you run, check:
1. make sure the name and type(in/out) of cell_pin are same with lib_verilog and lib_josim.
I have checked the name, to make sure pin_name correct.
for example:
`.a(), .b(), .q()` OK
`.a(), .b(), .c()` Error, pin_name not matched

*** if someone change libary, user must check pin_name.
*** recommand to write pins using same order with verilog library (same pin order and pin name).

2. make sure add '.print' and '.tran' command before using josim server. Such as:
```
.tran 0.2ps 1500ps 0ps
.print devi
.print devi Lip.XI3.Xand_03
```

3. The global output shouldn't flow to next component's pin_in. such as:
```
output cout;

and_bb and_05 (.q(cout), ... ) // ok
and_bb and_06 (.a(cout), ... ) < error,
```
output flow to someone's pin_in

4. Please write driver in config.txt, but if you want to change param of:
`xin1, xin2, din,`
please find them in code.

```
1251        # read config.txt from same dir and make a dict.
1252        config = {"xin1": "SIN (0 800mV 5GHz 100ps 0 )",
1253                  "xin2": "SIN (0 800mV 5GHz 150ps 0 )",
1254                  "din": "PWL ( 0ps 0mv 20ps 1200mV )"}
```

*** They are defined in code because user usually don't need to modify them.

5. A new function is added: force number of the cell-layers is **multiple of four**.
If you want to disable it, find in code.

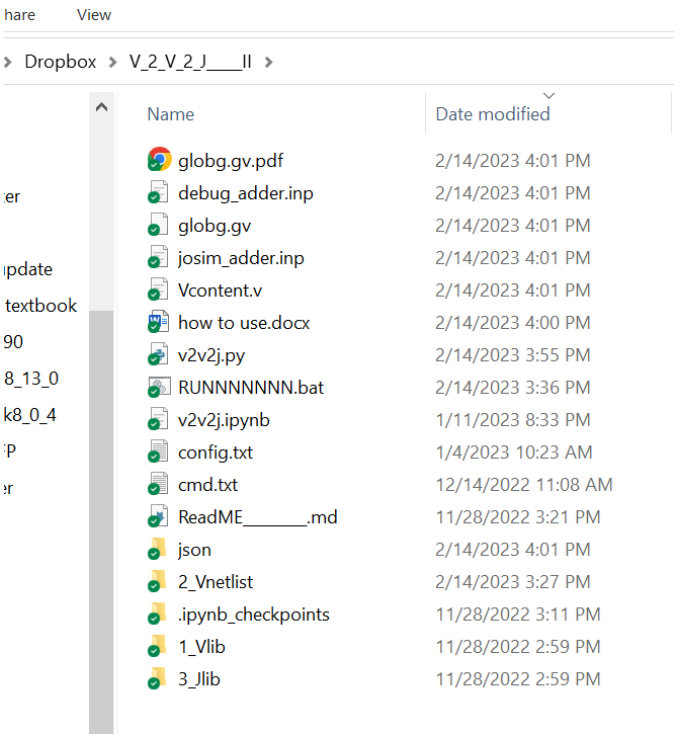
```
def pull_all_GlobOut_to_same_level(project_I, Four_=True):
    lis = []
    for i, node_i in enumerate(project_I["nodes_list"]):
        if node_i["object_name"] in project_I['output']:
            lis.append(node_i["level"])
    #print(lis)#[13,15,15,15]
    cell_lvs = max(lis)
    # check cell_lvs could %4 ==0
    if (cell_lvs%4 != 0):
        cell_lvs = ((cell_lvs//4)+1)*4
        if cell_lvs%4 != 0:
            print("Err, level%4")

    if Four_==True:
        for i, node_i in enumerate(project_I["nodes_list"]):
            if node_i["object_name"] in project_I['output']:
                node_i["level"] = cell_lvs # all to 15
        return project_I
    else:
        for i, node_i in enumerate(project_I["nodes_list"]):
            if node_i["object_name"] in project_I['output']:
                node_i["level"] = max(lis) # all to 15
        return project_I
```

Usage:

```
jsjs = get_level(jsjs)
jsjs = pull_all_GlobOut_to_same_level(jsjs, Four_=True)

jsjs = add_buffer(jsjs)

jsjs = make_connections(jsjs)
# ****** end here ***************************

jsjs = get_level(jsjs)
jsjs = pull_all_GlobOut_to_same_level(jsjs, Four_=True)
```

To run: you can double click **RUNNNNNNN.bat**
Or cmd: `python v2v2j.py`

When you complete run.
The folder will be like this:

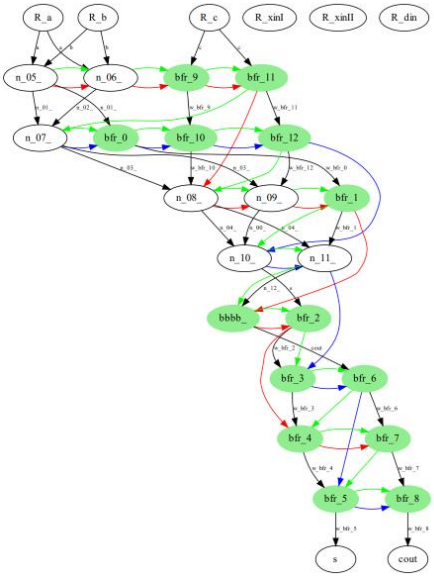


josim_adder.inp is the Josim file you will use.

Before using, don't forget to add add '.print' and '.tran' command!

globd.gv is a graphviz file. Describe all cells connections.

globd.gv.pdf is node connections figure: looks like:



Debug_adder.inp is for debug, all wires are shown in words, not number, for debug.

Vcontent.v is for debug, you can check the middle Verilog is correct.