

Project Phase 3 (MS3)

Done By:

Mohamed Taha-900172754

Omar Mahdy- 900160493

Youssef Beshir -900172772

Approach:

- Our Utility takes the verilog file name without extensions (i.e test1 -> python3 clk_gating_utility filename)
- The utility then parse the code using pyverilog code parser
- Calls the function parsingprocess(ast) which contains the logic of the utility
- Inside parsingprocess()
 - We created list newitems (to contain all the wires instance modules of our design, but we will appended and control which module to include and which to exclude, list L1, list L2, and some flags to be used
 - We then created a wire which will be the output of the clock gating cell and append it to our newitems
 - We then created the clock gating cell instance (skyl30 specific pdk)
 - Inside the food loop `for itemDeclaration in definition.items:` exists our logic of the code
 - We used a flag `flag2 = False` to make sure that we have only one instance of the ICG cell
 - We then used another flag `flag = True` to and set if to false when we have a mux so that we remove any mux instances and append in our newitems list only when the flag is true
 - Then we adjusted the inputs and outputs of the instances to match the design

- We searched for the clk portname in the flip flop and changed its argname to be the output of the ICG cell
- We then changed the input and output of the flip flop cells to match the inputs and outputs that were originally entering the muxes
- We then add the instances list to the AST items
- We then call the function that generate the output file and we add to its name the part "utility_out.gl.v"

How to use Utility:

- Open a terminal window in the same directory of the project files.
- Type: `python3 clk_gating_utility.py gatelevel_netlist_filename_without_extension`
- You will find the newly generated in the same directory of the old gate level netlist file with the same name + "utility_out.gl.v".

Progress:

- Till now we have provided the source code of our python utility that we have built in the previous milestone (Prototype) and tested it using a sample gate level netlist where the utility generated the new gate level where the old high power consuming structure is removed and replaced with the integrated clock gating (ICG) instance. Also, we have provided in the current milestone more test cases for our python utility which tests the utility using different sizes of input [1bit-2bits-3bits-4bits-32bits].
- In the next milestone we are planning to extend the functionality of our utility where it can accept and parse any cells other than the SkyWater130 cells and modify its structure to make it more power efficient. Moreover, we plan to provide an analysis to evaluate the power before and after to quantify the reduction.

Plans to evaluate the power before and after to quantify the reduction:

- The way we look to evaluate the power consumption before and after is we will run a testbench with the same test cases for the gate level netlist file ,once before and once after we run our utility program using pyverilog.
- Then we will instantiate two vcd files using iverliog to check the signals of both files.
- From checking the difference of the signals before and after we run our utility on it, we should observe the reduction in power as the signals after will represent the dynamic powered vcd file.