PROJECT: Equivalence Checking with Formality

In this project, you will learn how to use a commercial equivalence checker, called formality, to perform the equivalence checking on RTL design. Formality is an application that uses formal techniques to prove or disprove the functional equivalence of two designs or two technology libraries. For example, you can use Formality to compare a gate-level netlist to its register transfer level (RTL) source or to a modified version of that gate-level netlist. After the comparison, Formality reports whether the two designs or technology libraries are functionally equivalent. The equivalence checker is widely used in the industry to reduce design cycle for regression testing. In this project, we will only run the equivalence check on netlists or RTL level designs. You can explore other functionalities by yourself.

Software setup:

1. You will need to connect to the RedHat system remotely:

mo.ece.pdx.edu. If you are on campus, you can remote access through Ubuntu PC in the Intel lab. If you are off campus, check: https://cat.pdx.edu/services/network/vpn-services/ to install PSU VPN on your system. Then install a remote access software support x-server such as MobaXterm or putty with xming.

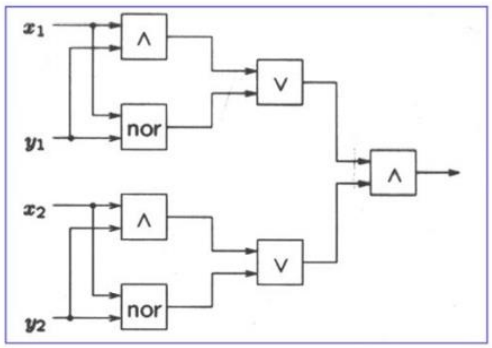
2. Login to mo.ece.pdx.edu with your ECE username and password.
3. Run addpkg and select DC, formality, and then re-log in the system.

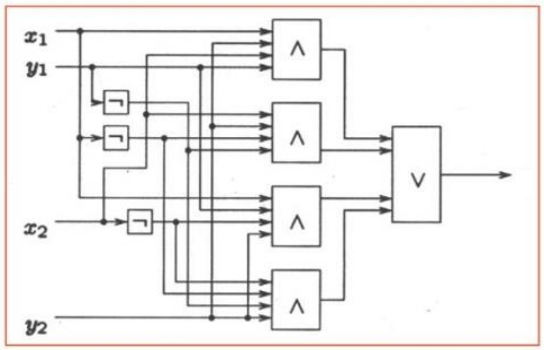
```
    ts2017
[X] synopsys-designc  SYNOPSYS Design Compiler (G-2012.06)
    ompiler
[ ] synopsys-dftcomp  SYNOPSYS DFT Compiler, with Tetramax Overlay (2016)
    iler
[X] synopsys-formali  SYNOPSYS Formality (2017)
    ty
[X] synopsys-formali  SYNOPSYS Formality ESP (2017)
    ty_esp
```

4. Create a workstation directory for your formality project in your home directory, place your Verilog file you want to check in the directory and run fm_shell form there. You can also run fm_shell -gui in GUI mode, so you can just select your files instead of using commands.
5. Read the tutorial to practice some basic commands.
6. User Manuel can be found at D2L reading materials.

Task 1: Describe the following two circuits C1 and C2 in Verilog. Use the formality to check if they are equivalence. You need to use the same names on the corresponding inputs and outputs.

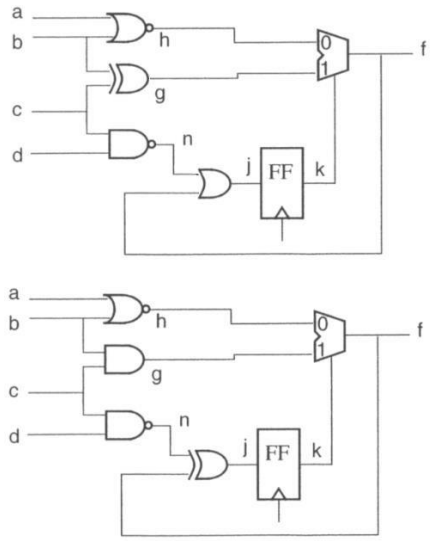
Task 1.1:





Task 2: Use the formality to check if two sequential circuits are equivalent.

Task 2.1: Perform the equivalence check on the following two sequential circuits with formality.



Note:

1. For those who have not used any Synopsys tools before, you might have the license problem. If you see an error message related to the license or path issue while running formality, you can use the addpkg command, then add formality 2017. This will set up the Synopsys license path for you.
2. The report should show the screenshots of the result from the formality.