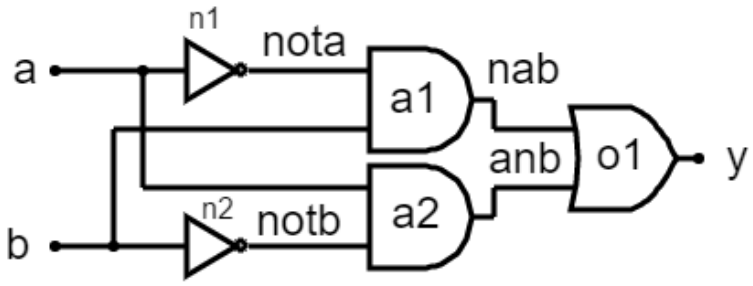
CSULB
CECS225

Lab

Objectives: Model a logic circuit using gate level primitives to learn port mapping for interconnecting modules.
Procedure: In this lab a logic circuit will be modeled using **built in library module components for the and gate, the or gate, and the not gate (inverter)**.

//write the function using assign
//write the function using primitive gates



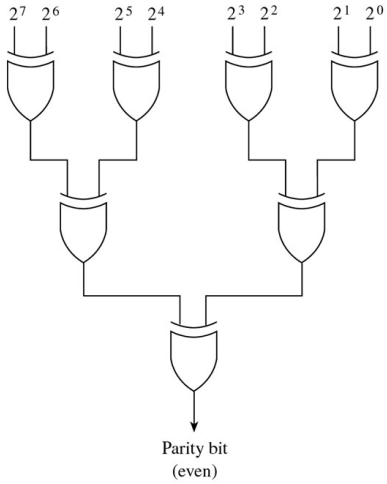
Notice the relationship between the logic circuit and the Verilog module after it.
Internal wires must be declared inside the module as local variables. Inputs and the output are considered local variables. The port list for each logic gate has parameters which follow this order:
(output, input1, input2, … inputN)

As it can be seen the Verilog module for xorGate is a list of all the connections depicted in the logic diagram above.

Then use the design to build the comparator using the one bit comparator design you built.

Note the way that reserved keywords are used to place the logic gates. Each logic gate also has a unique identifier which is followed by a list of connections known as a port list. Multiple instances of the same type of module (or the same type of gate) can be coded conveniently by using a comma after a port list before providing a unique identifier for the next instance of the gate/module followed by its port list and the last instance is followed by a semicolon.

Your task is to model the following circuit using **gate level primitives** to build an **ex-or gate** and then use "**Structural Modeling – Hierarchy**" given the following Verilog module skeleton.



Complete the testbench code, generate your timing diagram, your outputs and the and checking results with inline code.

Record yourself explaining the code and generating your output. Upload your video to YouTube. Paste your link to the video with your your design and testbench codes. Take screenshots of your outputs and upload as one pdf file.