Experiment#6

Title: -

Implementation of De Morgan's law and Distributive law

Objective: -

- To get familiar with Verilog
- To get familiar with SynaptiCAD verilogger software
- To implement Demorgan's and Distributive law

Tools required: -

- SynaptiCAD verilogger Software

Theory: -

What is Verilog and why we use it?

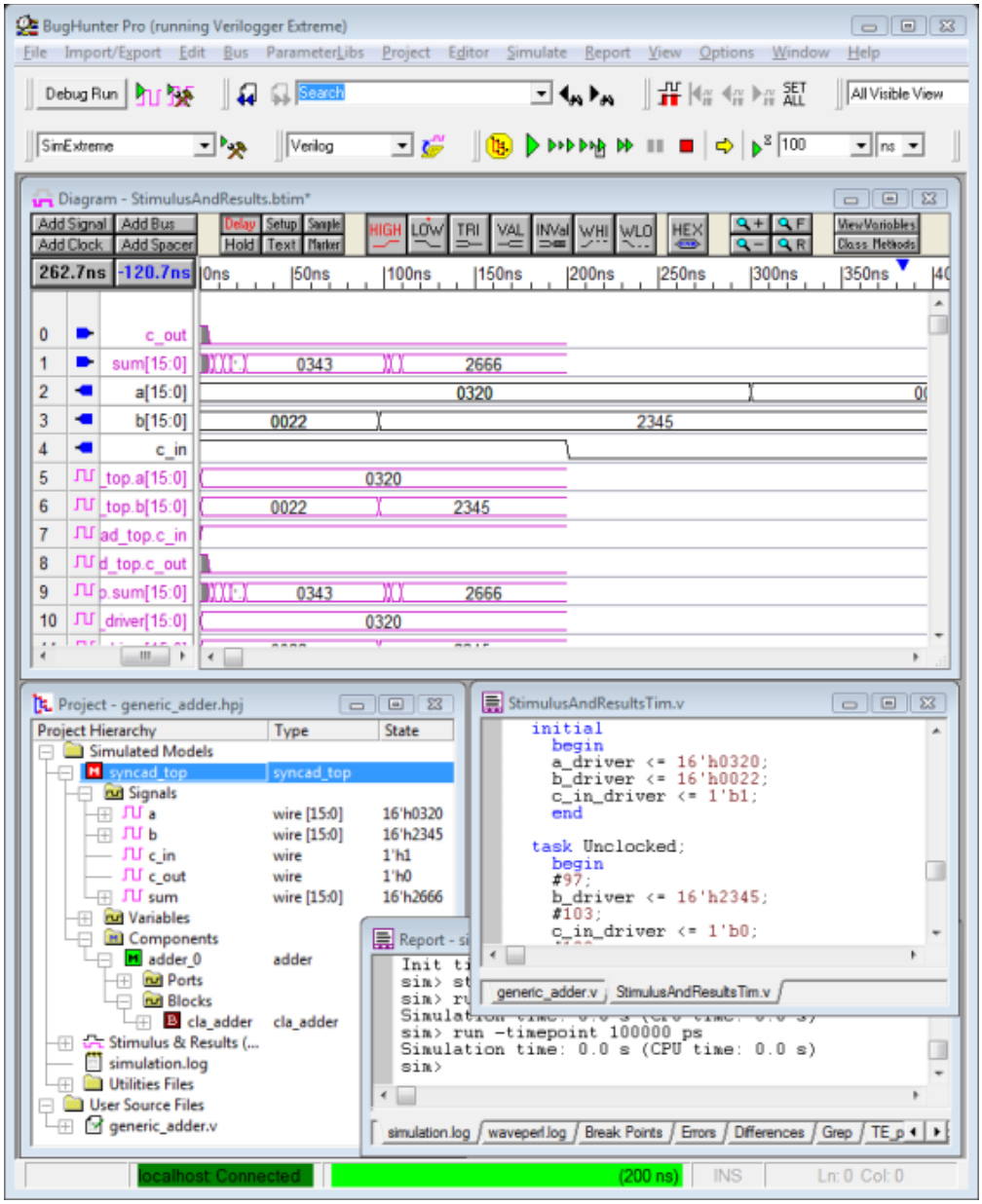
Verilog is a HDL (Hardware Description Language). It is used to model and simulate digital electronic circuits. Once a design is simulated, tested and ready for 'tape-out' to the fab, it can be synthesized to produce gate level designs that are then translated to physical design. In some ways, you can think of Verilog HDL as the code that represents the highest level abstraction of electronic circuits. Verilog is a hardware description language. It is different from general purpose programming languages in that it is specifically used to model hardware. So in Verilog you have the ability to specify registers, wires, gates, clock, etc. It is very useful when you have hardware specs on paper, and you want to simulate and test it first before synthesizing the circuit, thus saving time and money. The language is also pretty simple, only requires knowledge of digital logic, and it has syntax similar to C. The "Veri" part of the name is short for verification, and Verilog was designed for simulating digital hardware to verify that it operated correctly prior to fabrication. Some of the language constructs pertain to board-level logic (e.g. tri-state buses), but it is mostly known as a digital IC modeling language. It was the "sign off" simulation language for chip design for many years, and still is in some quarters. System Verilog is a pure extension of Verilog, but has not appreciably improved the sign-off level capabilities of the language, it mostly added test-bench level constructs (that could have been done in C/C++ or other OO languages), and sequenced assertions which help with verification.

SynaptiCAD Verilogger:

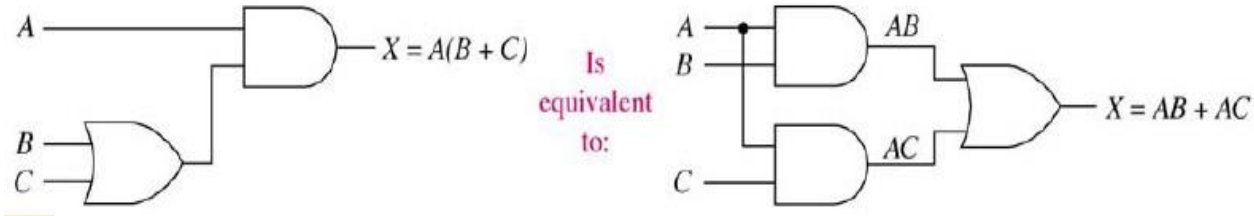
VeriLogger, by SynaptiCAD is a complete design and verification environment for ASIC and FPGA designers. It contains a new type of Verilog simulation environment that combines all the features of a traditional Verilog simulator with the most powerful graphical test vector generator on the planet. Model testing is so fast in VeriLogger Pro that you can perform true bottom-up testing of every model in your design, a critical step often skipped in the race to market. Test vectors can be imported or exported from HP logic analyzers, pattern generators, and 3rd party VHDL, Verilog, and SPICE simulators for reuse. Simulation features include waveform viewing, optimized gate-level simulation, single-step debugging, point-and-click breakpoints, hierarchical browser for project management, and batch execution.

VeriLogger Screen Shot

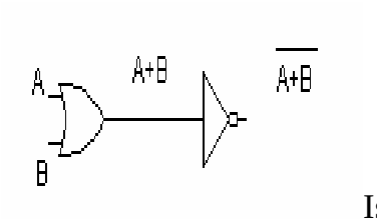
Take a look at everything available to you in the VeriLogger Program:



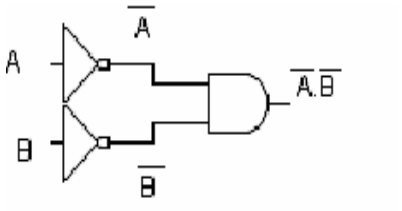
Distributive law

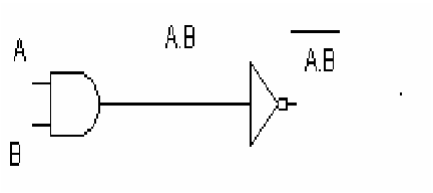


DeMorgan's law

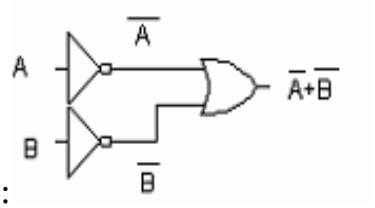


Is equivalent to**:**





Is equivalent to:



Task:

1. Implementation of distributive's law using Verilog
    a. Draw the circuit diagram
    b. Paste the code or screenshot of the code
    c. Paste the screenshot of timing diagram