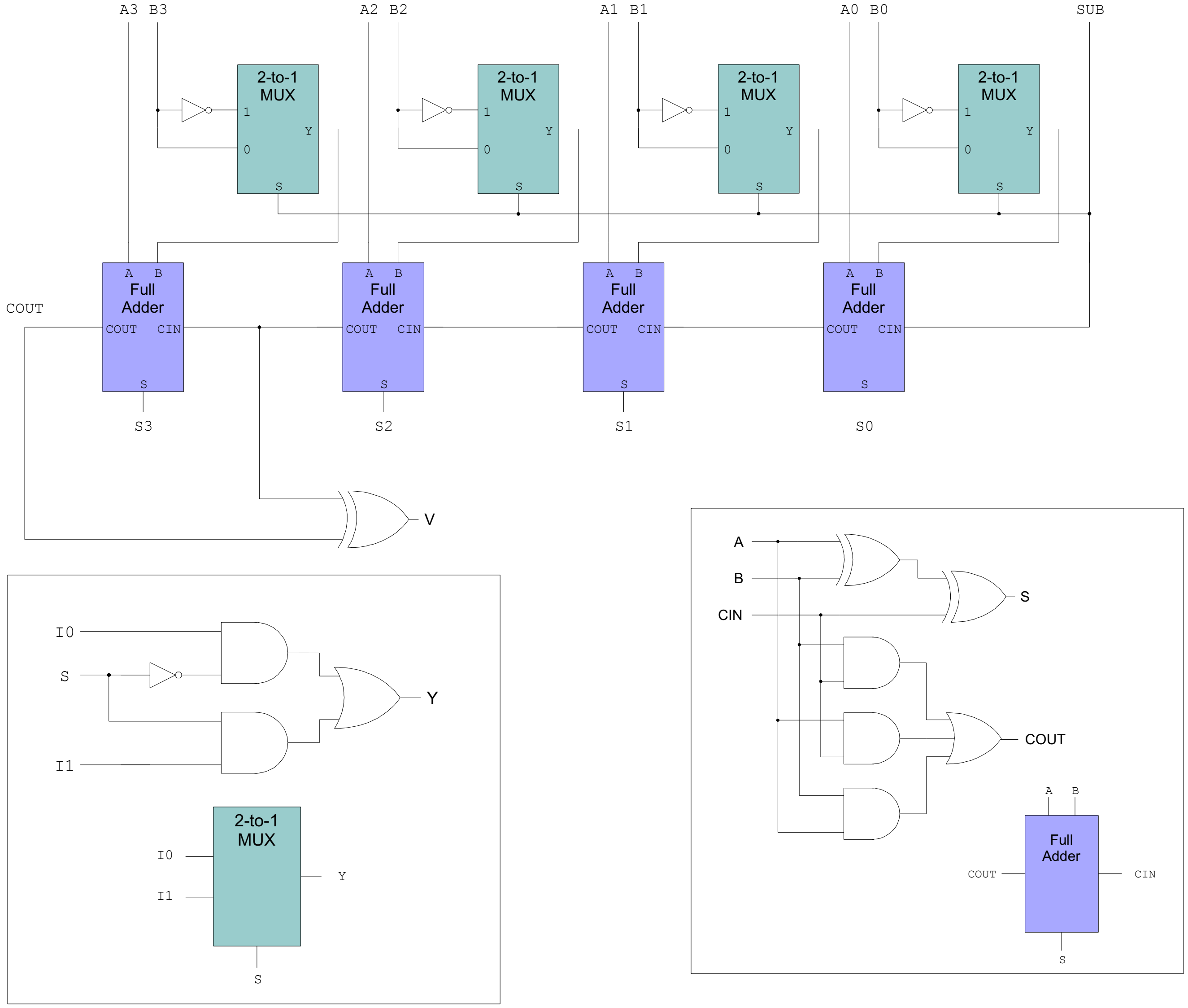
CSE 341
Fall 2025
Project #2

Due: 11:59 PM on Sunday, November 16, 2025
It is important that you start early and do NOT procrastinate. To encourage this, help will not be available for questions related to Verilog after Friday, November 7, 2025. ***START EARLY!***

Project Description

Your task is to implement a 32-bit ripple carry adder/subtractor with overflow (V) using Verilog HDL and analyze it by simulating it in ICARUS Verilog on *timberlake*. A functional (zero delay) simulation should be used to verify the circuits works properly and a unit gate delay model simulation should be used to evaluate performance. ***Structural*** (not behavioral) Verilog ***must*** be used. The circuit that should be implemented is what is in the lecture notes on Verilog and digital logic. Circuit diagrams are shown below for your convenience. Note the 4-bit adder/subtractor is shown for brevity, but you are implementing a 32-bit.



Simulation & Analysis

A written report must be submitted which includes functional (zero delay) simulation results proving that the adder/subtractor works properly along with a detailed delay analysis. The functional simulation must include five simulation runs that provide compelling evidence that your circuit adds properly, five runs proving that your circuit subtracts properly, five runs where overflow (V) occurs, and five additional runs where overflow does not occur. Compelling evidence constitutes testing the adder/subtractor over the entire range of inputs. Testing cases such as 1+2, 3+1, 4+0, 6+2, and so on doesn't provide compelling evidence as they do not test the upper order bits of the adder/subtractor. Each test case should be accompanied by comments that describe why the test case was chosen. Once you have proved that your adder works from a functional perspective, you can then explore various scenarios under the unit gate delay model, as outlined in the following list. This will help you gain a better understanding of how delays work in real circuits. The required elements of your report are outlined below.

- Functional simulation results, as detailed below. When submitting your Verilog source files for the project, the Verilog code associated with this functional simulation should be called *functional_simulation.v*).
  - Five data sets proving your circuit adds properly
  - Five data sets proving your circuit subtracts properly
  - Five data sets proving your circuit properly flags signed overflow (V) when it occurs, setting the output V to a one
  - Five data sets proving your circuit properly generates an overflow output (V) of zero when overflow does not occur
- Critical path of the adder/subtractor under the unit gate delay model.
- Critical path delay of the adder/subtractor under the unit gate delay model.
- Show a specific example of temporal delay when your circuit is simulated under the unit gate delay model. In other words, show how delay is dependent upon the current inputs, AND the previous set of inputs.
- Unit gate delay simulation results for the adder/subtractor, showing the delay for each of the 20 input patterns used in the functional simulation described above.
  - At least one must result in a delay that is greater than 50% of the critical path delay.
  - At least one must result in a delay that is less than 50% of the critical path delay.
  - When submitting your Verilog source files for the project, the Verilog code associated with this functional simulation should be called *unit_gate_delay_simulation.v*).
- The average delay of the unit gate simulation over 5,000 input patterns.
- Explanation of how you obtain the average delay simulation results in the previous step. In other words, what was your method for setting up the Verilog source file and analyzing the results.
- Analysis of the average delay over the 5,000 inputs patterns, indicating if the delay less than or equal to the critical path delay and why.
- A specific example showing a glitch in your circuit.

Real World Analysis

Case #1

You provided simulation with compelling evidence that your circuit works properly. Consider the importance of this step in terms of a real world example. Circuits, like the one you implemented, are used in embedded systems that can have deadly consequences if designed improperly. Discuss the best way to test your circuit to maximize the likelihood that it works properly along with the impact if it is not functioning properly. In doing so, consider the case whereby your circuit is to be utilized in the embedded systems that control a passenger aircraft.

Case #2

Now that you have witnessed first hand how delays propagate through a circuit, consider that Apple transitioned from the Intel 80x86 based architecture to the ARM based M1/M2/M3/M4 processor. The M1/M2/M3/M4 processor is a system-on-chip (SOC) architecture, reducing the physical size of the hardware by incorporating the circuitry onto a single chip. Research the impact that these architectures have on power consumption and the environment. Analyze and compare the two architectures based on your research. Present your argument on which is better.

Submission

Submit your commented Verilog code, along with a PDF of your report on using the submit command (*submit_cse341 project_3.pdf functional_simulation.v unit_gate_delay_simulation.v*) on *turing.cse.buffalo.edu*. Your Verilog code will already be on *turing* since that is where you are running your simulations. Up will have to upload the PDF of your project report to *turing*, and once uploaded, submit your project.