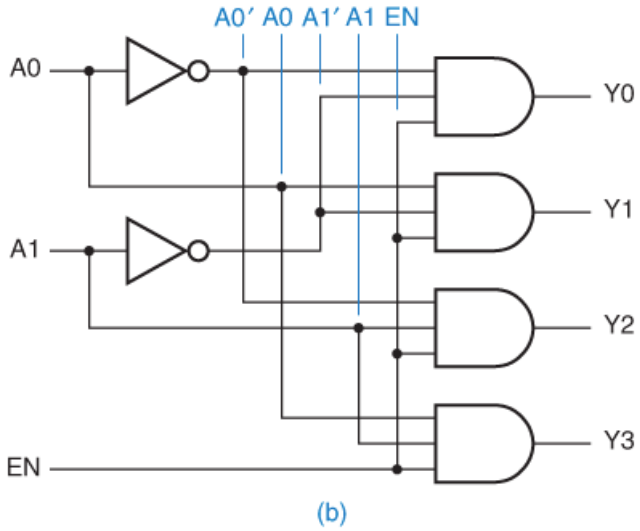
Assignment #3
EECS 303: Advanced Digital Logic Design

Problem 1. (20 Points):

Write RTL style Verilog module for the combinational circuit in the figure shown below.

Write a testbench and simulate the Verilog code using Excelium for all possible values of the three inputs. Include screenshot of the simulation output.



(b)

Problem 2 (20 Points) Write RTL-style Verilog module corresponding to the full adder circuit in Figure 2.1. Then, instantiate multiple copies of it to create a structural model for a 4-bit ripple carry adder structure of Figure 2.2. Create a testbench for the adder, apply four different input combinations and attach screenshots of the simulation showing correct behavior.

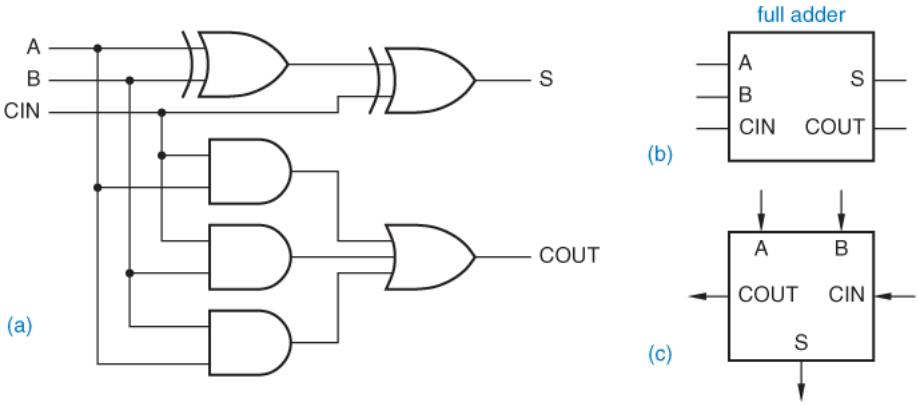


Figure 2-1 Full adder: (a) gate-level logic diagram; (b) logic symbol; (c) alternate logic symbol suitable for cascading.

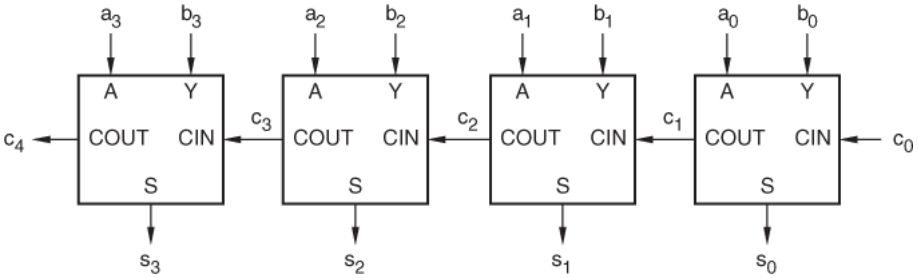
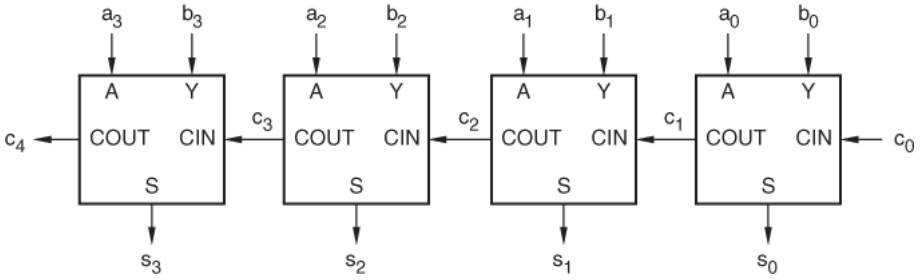


Figure 2-2 A 4-bit ripple adder.

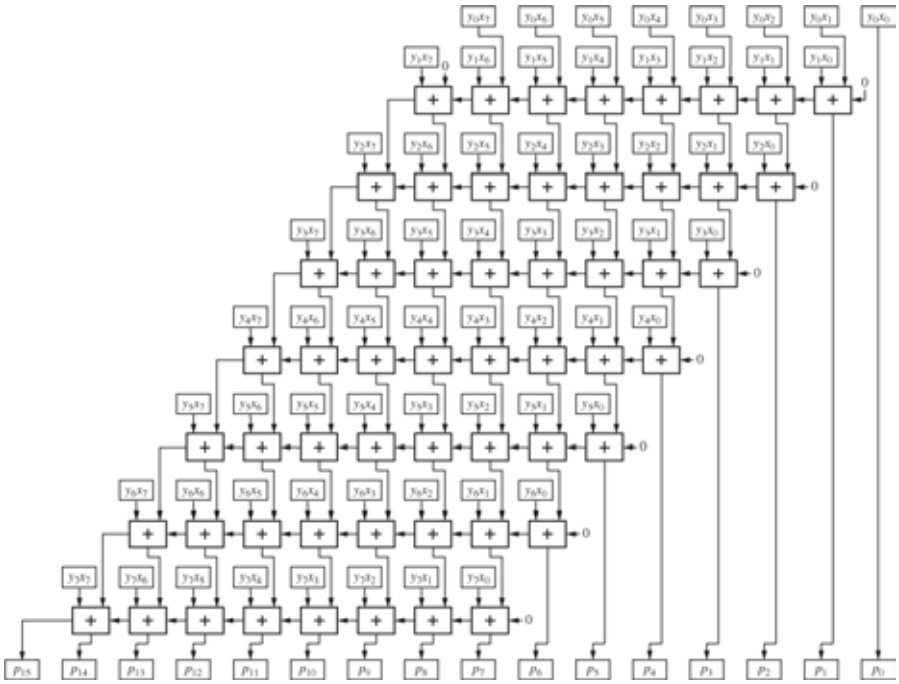
Problem 3. Arithmetic Circuits (20 Points)

a. Assume gates such as NOT, NAND, NOR, AND, OR have a delay of 1 unit and an XOR or XNOR gate has a delay of 3 units. What is the worst-case delay from any input to any sum output for the 4-bit ripple adder in the Figure? What is the worst-case delay to the carry output c_4?



4-bit ripple adder.

b. Determine the worst-case propagation delay of the multiplier in Figure below, assuming that the propagation delay from any full-adder input to its sum output is twice as long as the delay to the carry output. Repeat, assuming the opposite relationship. If you were designing the adder cell from scratch, which path would you favor with the shortest delay?



Interconnections for an 8 × 8 combinational multiplier.

Problem 4. Arithmetic Circuits (10 Points)

What is the delay through the critical path in the 8-bit Kogge-Stone adder? Assume that it gets implemented with 2-input simple logic gates (and, or, exor, not) and that a gate delay equals 1 unit of time. Trace the path through the adder by listing the number and type of each gate.

Problem 5. Arithmetic Circuits (30 Points)

Write Dataflow-style Verilog module for a 3x3 (i.e., each input is 3 bits) combinational multiplier and simulate with all possible input combinations. Use explicit **assign** statements to define each partial product calculation and you can use **assign** statements with arithmetic addition to accumulate those partial products.