

Warm-Up: Adders

Prepared by Mark on February 12, 2025

Instructor's Handout

This handout contains solutions and notes.

Recompile without solutions before distributing.

Problem 1:

Fill the following binary addition table.

Hint: s is “sum,” c is “carry”

| a | b | s | c |
|-----|-----|-----|-----|
| 0 | 0 | ? | ? |
| 0 | 1 | ? | ? |
| 1 | 0 | ? | ? |
| 1 | 1 | ? | ? |

Problem 2:

Draw a logic circuit that satisfies the above table.

This is called a *half adder*.

Hint: You should need exactly two gates.

Solution:

$$s = a \text{ xor } b$$

$$c = a \text{ and } b$$

Definition 3:

A *full adder* is similar to a half adder, but it has an extra input:

a full adder takes a , b , and c_{in} , and produces s and c_{out} .

Hint: c_{in} is “carry in”

Problem 4:

Use two half adders to construct a full adder.

Solution:

$$s_1, c_1 = \text{HA}(a, b)$$

$$s_2, c_2 = \text{HA}(s_1, c_{\text{in}})$$

$$s_{\text{out}} = s_2$$

$$c_{\text{out}} = \text{OR}(c_1, c_2)$$

Of course, the class should just draw the circuit.

Problem 5:

How can we add two four-bit binary numbers using the full adder?

We want a four-bit output sum and a one-bit c_{out} .

Problem 6:

Say that all basic logic gates need $1u$ of time to fully switch states.

Note: This is called *gate delay*

How much time does a full adder need to fully switch states?

How about your circuit from Problem 5?

Problem 7: Bonus

Design a faster solution to Problem 5.