**Exercise 3.29.**

Another way to construct an or-gate is as a compound digital logic device, built from and-gates and inverters. Define a procedure `or-gate` that accomplishes this. What is the delay time of the or-gate in terms of `and-gate-delay` and `inverter-delay`?

**Answer.**

Consider the law of De Morgan, which shows different ways for expressing the same logical function:

$$\begin{aligned} A_1 \vee A_2 &= \overline{\overline{A_1 \vee A_2}} \\ &= \overline{\overline{A_1} \wedge \overline{A_2}} \end{aligned}$$

To erecting an or-gate of two input signal $A_1$ and $A_2$ from and-gates and inverters, we first invert each of them respectively, then compute the `logical-and` of their output and obain the designated `logical-or` by one more inverting. Figure 1 shows such an or-gate build from and-gates and inverters.

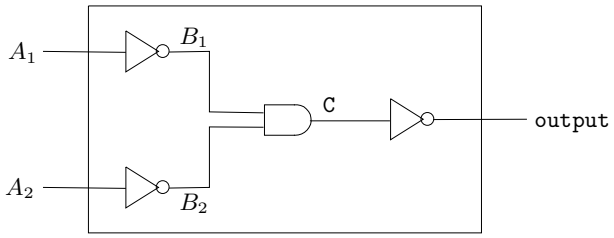


**Figure 1.** An or-gate build from and-gates and inverters.

We can also express this idea in Lisp:

```
(define (or-gate a1 a2 output)
  (let ((b1 (make-wire))
        (b2 (make-wire))
        (c (make-wire)))
    (inverter a1 b1)
    (inverter a2 b2)
    (and-gate b1 b2 c)
    (inverter c output)
    'ok))
```

Notice that in this implementation, although the or-gate is built from three inverters and an and-gate, the first two inverters propogate their output to the and-gate parallelly, rather than serially. This indicates that the and-gate waited merely one `inverter-delay` until it began to proceed the signal. Hence, the delay time of the or-gate here turn out to be:

$$\texttt{or-gate-delay} = 2 \times \texttt{inverter-delay} + \texttt{and-gate-delay}$$