0.1 Logic Gate Array that Produces an Arbitrarily Chosen Output

Suppose that our task is to design a logic gate array that has three binary inputs — labelled **A**, **B** and **C** — and that produces some particular set of outputs, **F**. The truth table that we'll take as an example is shown in Table 0.3. (The output values were just picked out of thin air and don't represent anything in particular.)

Note that we've written the binary inputs on the left in a very regular way: if regarded as being three-digit binary numbers, their eight values would increase in steps of one from zero at the top to seven at the bottom. Writing them in this structured way helps us avoid leaving out any possible values.

Here's a set of steps that is guaranteed to produce a logic gate array that behaves the same as the truth table that you start with:

Step 1: Use an OR gate as the output gate. Provide this OR gate with as many inputs as there are **1**s in the output (the F column). In our example, the output contains four **1**s, and so the output OR gate must have four inputs, corresponding to the third, fourth, sixth, and eighth rows in the truth table above.

Step 2: Set up a general-purpose circuit on the input side, at the left, that will take in the inputs (**A**, **B**, and **C**) and that will provide internally both the inputs (**A**, **B**, and **C**) and the negations of the inputs ($\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$, and $\overline{\mathbf{C}}$) in easily accessed columns. NOT gates are used to form the negations ($\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$, and $\overline{\mathbf{C}}$) from the inputs (**A**, **B**, and **C**).

Table 0.3 What circuit of logic gates could produce these (arbitrarily chosen) outputs F in response to inputs A, B and C? See text for method and Figure 0.1 for result.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

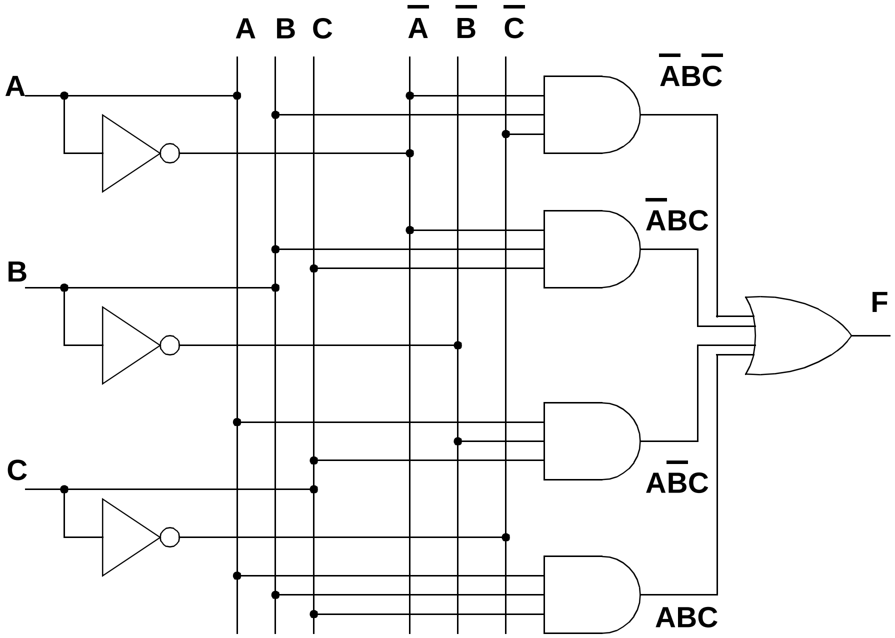

Figure 0.1 Example of general purpose circuit to implement the truth table of Table 0.3. (This solution is NOT minimized.)

Step 3: Use an AND gate with as many inputs as the circuit (three in this case) to compute each row of the truth table that produces a **1**. Connect the output of each AND gate to one of the inputs of the OR gate. For example, the first set of inputs that would produce a **1** output is the third row of the arbitrarily chosen truth table. To get a **1** output from the OR gate when the inputs A, B, and C have values **0**, **1**, and **0** (the third row of the truth table), we would connect $\overline{A}$, B, and $\overline{C}$ to a three-input AND gate as shown in Figure 0.1.

Prentice-Hall, Inc.
Upper Saddle River, New Jersey 07458