

Supplement to

Logic and Computer
Design Fundamentals
4th Edition¹

ADVANCED TECHNOLOGY MAPPING

Selected topics not covered in the fourth edition of *Logic and Computer Design Fundamentals* are provided here for optional coverage and for self-study. This material fits well with the desired coverage in some programs but may not fit within others due to time constraints or local preferences. This supplement uses material from the third edition of *Logic and Computer Design Fundamentals* and is intended to be used as a selected topic before Section 6-8 or following Section 3-3 of the fourth edition. Material covered includes VLSI circuit design approaches, cell libraries, and a general procedure for mapping combinational logic circuits to cell library-based implementations. This material is appropriate for students in programs in electrical engineering courses where implementation is of importance.

VLSI Design Approaches

There are three primary ways of designing VLSI circuits that serve as foundations for technology mapping. In *full custom design*, an entire design of the chip, down to the smallest detail of the layout, is performed. Since this process employs manual technology mapping, and, consequently, is very expensive, custom design can be justified only for dense, fast ICs that are likely to be sold in sizable quantities.

A closely related technique is *standard cell design*, in which large parts of the design have been performed ahead of time or, possibly, used in previous designs. The predesigned parts are connected to form the IC design. This intermediate-cost methodology gives lower density and lower performance than full custom design.

The third approach to VLSI design is the use of a *gate array*. A gate array uses a rectangular pattern of gates fabricated in silicon. This pattern is repeated thousands of times, so that the entire chip contains identical gates. Depending on

¹© Pearson Education 2008. All rights reserved.

the technology used, pattern arrays of 1000 to millions of gates can be fabricated within a single IC. The application of a gate array requires that the design specify how the gates are interconnected and how the interconnections are routed. Many steps of the fabrication process are common and independent of the final logic function. These steps are economical, since they can be used for numerous different designs. In order to customize the gate array to the particular design, additional fabrication steps are required to interconnect the gates. Due to the commonality of fabrication steps and ability to share the results of these steps with many different designs, this is the lowest cost method among the fixed implementation technologies.

Cells and Cell Libraries

For standard cell and gate array technologies, circuits are constructed by interconnecting cells. The collection of cells available for a given implementation technology is called a *cell library*. In order to design in terms of a cell library, it is necessary to characterize each of the cells (i.e., provide a detailed specification of the cells for use by the designer). A library of characterized cells provides a foundation for the technology mapping of circuits. Coupled with the library is a technology mapping procedure. In this section, we consider technology mapping procedures for cell libraries consisting of multiple gate types.² Technology mapping may focus on a number of the dimensions of the design space, particularly on cost and performance. For simplicity, our procedures focus only on optimizing cost. Further, these procedures are rudimentary versions of technology mapping algorithms used in computer-aided design tools and are suitable for manual application to only the simplest of circuits. Nevertheless, they give us some insight into how a design using AND gates, OR gates, and inverters can be automatically transformed into cost-effective designs using cell types supported by available implementation technologies.

CELL SPECIFICATION Specifications for cells used in standard cell and gate array designs typically have many components. Typical components include the following:

1. A schematic or logic diagram for the function of the cell.
2. A specification of the area the cell occupies, often normalized to the area of a small cell such as that of a minimum area inverter.
3. The input loading, in standard loads, that each input of a cell presents to the output driving it.
4. Delays from each input of a cell to each output of a cell (if a path from the input to output exists), including the effect of the number of standard loads driven by the output.
5. One or more templates for the cell for use in performing technology mapping.
6. One or more HDL models for the cell.

²Technology mapping for NAND or NOR gates is covered in Section 3-3 of the text.

If the tools used provide automated layout, then the following additional components are also included in the specification:

7. An integrated circuit layout for the cell.
8. A floorplan layout showing the locations of the inputs, outputs and power and ground connections for the cell for use during the cell interconnection process.

The first five components listed are included in a simple technology library of cells in the next subsection. Some of these components are discussed in more detail.

LIBRARIES The cells for a particular design technology are organized into one or more libraries. A *library* is a collection of cell specifications. A circuit that initially consists of AND, OR and NOT gates is converted by technology mapping to one that uses only cells from the applicable libraries. A very small technology library is described in Table 1. This library contains primitive inverting gates with fan-ins up to four and a single AOI circuit.

The first column of the table contains a descriptive name for the cell and the second column contains the cell schematic. The third column contains the area of the cell normalized to the area of a minimum inverter. Area can be used as a very simple measure of the cost of the cell. The next column gives the typical load that a cell input places on the gate driving it. The load values are normalized to a quantity called a standard load which in this case is the capacitance presented to the driving circuit by the input of an inverter. In the case of the cells given, the input loads are almost all the same. The fifth column gives a simple linear equation for calculating the typical input-to-output delay for the cell. The variable SL is the sum of all of the standard loads presented by the inputs of cells driven by the cell output. It may also contain an estimate, in standard loads, of the capacitance of the wiring connecting the cell output to the inputs of other cells. This equation illustrates the notion that cell delays consist of some fixed delay, plus a delay that is dependent upon the capacitance loading of the cell as represented by SL. Cell delay calculation is illustrated in Example 1.

• **EXAMPLE 1 Calculation of Cell Delay**

This example illustrates the effect of loading on cell delay. A 2NAND output drives the following cells: an inverter, a 4NAND, and a 4NOR. The sum of the standard loads in this case is






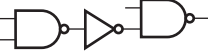

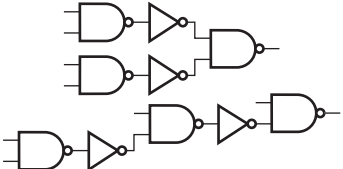

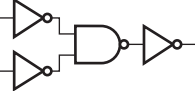

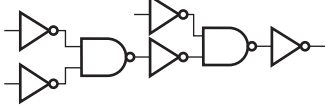

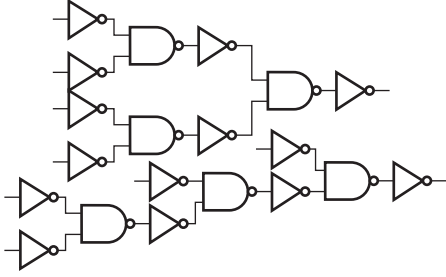
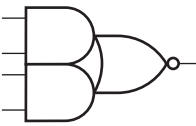
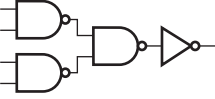
$$SL = 1.00 + 0.95 + 0.80 = 2.75$$

With this value, the delay of the 2NAND driving the cells specified is

$$t_p = 0.05 + 0.014 \times 2.75 = 0.089 \text{ ns}$$

The final column of the table gives templates for the cell function that use only basic functions as components. In this case, the basic functions are a 2-input NAND gate and an inverter. Use of these basic function templates provides a way of representing each cell function in a “standard” form. As illustrated by the 4-input NAND and NOR cells, the basic function template for a cell is not necessar-

TABLE 1
Example Cell Library for Technology Mapping

Cell Name	Cell Schematic	Normalized Area	Typical Input Load	Typical Input-to-Output Delay	Basic Function Templates
Inverter		1.00	1.00	$0.04 + 0.012 \times SL$	
2NAND		1.25	1.00	$0.05 + 0.014 \times SL$	
3NAND		1.50	1.00	$0.06 + 0.017 \times SL$	
4NAND		2.00	0.95	$0.07 + 0.021 \times SL$	
2NOR		1.25	1.00	$0.06 + 0.018 \times SL$	
3NOR		2.00	0.95	$0.15 + 0.012 \times SL$	
4NOR		3.25	0.80	$0.17 + 0.012 \times SL$	
2-2 AOI		2.25	0.95	$0.07 + 0.019 \times SL$	

ily unique. It should be noted that these diagrams represent only a netlist, not actual location, orientation, or interconnect layout. For example, consider the template for the 3NAND. If the left NAND and the following inverter were connected to the top input of the right NAND, instead of to its bottom input, the template would be unchanged. The value of these templates will become apparent in the next section on mapping techniques.

General Technology Mapping Procedure

The following procedure handles mapping for a general library of single-output combinational cells:

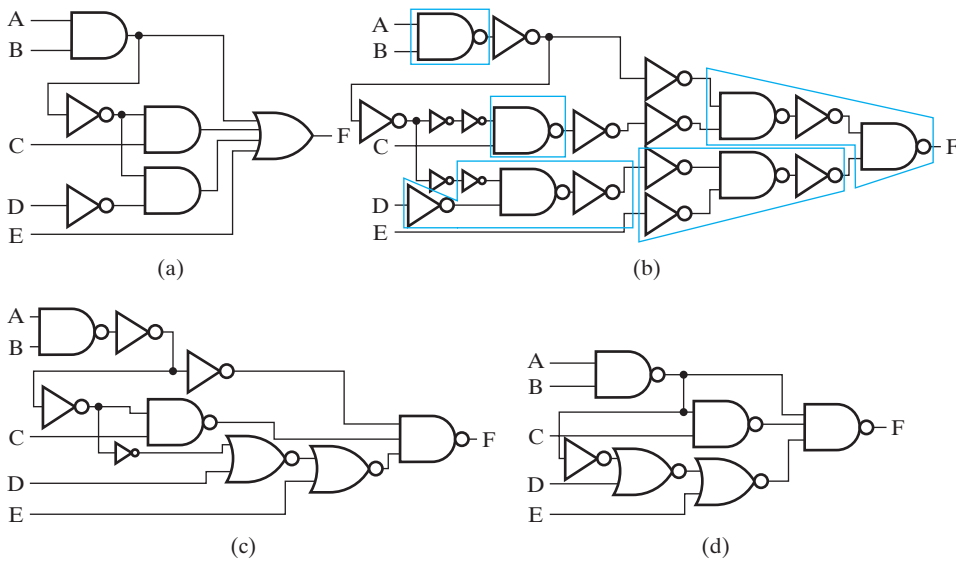
1. Replace each AND and OR gate with an optimum equivalent circuit consisting only of 2-input NAND gates and inverters.
2. In each line in the circuit attached to a circuit input, a NAND gate input, a NAND gate output, or a circuit output in which no inverter appears, insert a serial pair of inverters.
3. Perform a replacement of connections of NAND gates and inverters by the available library cells such that the gate input cost which results within fan-out free subcircuits is optimized. A *fan-out free subcircuit* is a circuit in which each gate output drives a single gate input. (This step is not covered here in detail due to its complexity). The templates shown in the right column of Table 1 are used to match connections of NAND gates and inverters to available library cells.)
4. (a) Without changing the logic function, “push” all inverters, lying between (i) a circuit input or a driving gate output and (ii) the driven gate inputs, toward the driven gate inputs. Cancel pairs of inverters in series whenever possible during this step. (b) Replace inverters in parallel with a single inverter that drives all of the outputs of the parallel inverters. (c) Repeat (a) and (b) until there is at most one inverter between the circuit input or driving gate output and the attached driven gate inputs.

This procedure is one of the foundations for technology mapping in commercial synthesis tools. The intermediate replacement of the initial circuit gates with only 2-input NAND gates and inverters breaks the circuit up into small pieces in order to provide the maximum flexibility in mapping cells to achieve an optimized result. Example 2 shows an implementation approach using a small cell library.

• EXAMPLE 2 Mapping with a Small Mixed Cell Library

$$F = AB + (\overline{AB})C + (\overline{AB})\overline{D} + E$$

with a cell library containing a 2-input NAND gate, 3-input NAND gate, a 2-input NOR gate, and an inverter. The AND, OR, inverter implementation is given in Figure 1(a). In Figure 1(b), steps 1 and 2 of the procedure have been applied. Each AND gate and each OR gate has been replaced with its equivalent circuit made up of 2-input NAND gates and inverters. Pairs of inverters have been added to the



□ **FIGURE 1**
Solution to Example 2

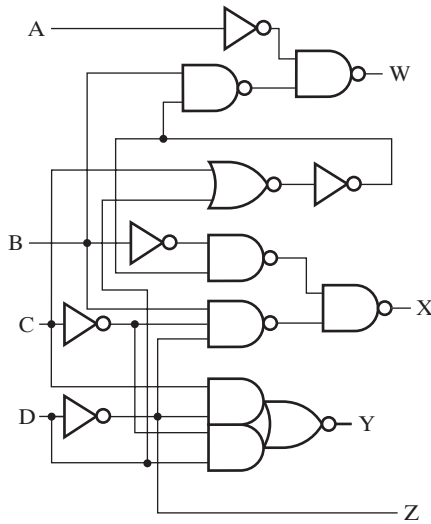
internal lines without inverters. Due to lack of space, the pairs of inverters on the inputs and outputs are not shown. Application of step 3 results in the mapping to the cells from the cell library as shown in Figure 1(c). The blue outlines enclose connections of NAND gates and inverters, each of which is to be replaced by an available cell using the templates in Table 1. In this case, all of the available cells have been used at least once. Application of step 4 cancels out three of the inverters, giving the final mapped circuit in Figure 1(d). The solution has a gate input cost of 12. •

To provide continuity with examples in the text, the following example shows the mapping of the BCD-to-Excess-3 Code Converter for the cell library in Table 1.

• **EXAMPLE 3 Technology Mapping for BCD-to-Excess-3 Code Converter**

The final result of the technology mapping for the BCD-to-Excess-3 Code Converter is given in Figure 2. The original AND, OR, inverter logic diagram appears in Figure 3-2 of the text, and the cell library used is given in Table 1. The optimization has resulted in the use of the following cells from that library: inverters, 2-input NANDs, a 2-input NOR, and a 2-2 AOI. •

The gate input cost of the mapped circuit in Example 3 is 22. The NAND gate mapping in Figure 3-9 in the text has a gate input cost of 21. So the expanded cell library in this case did not yield a better result. Further, the optimization did not yield the simpler solution in Figure 3-9. The optimization procedure, aside from locally minimizing inverters, works separately on the various parts of the circuit. These parts are separated by gate fan-outs in the original AND-OR circuit. The selection of these points during optimization can affect the optimality of the final



□ **FIGURE 2**
Technology Mapping Example: BCD-to-Excess-3 Code Converter

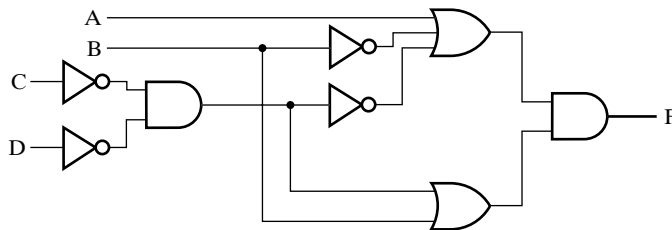
result. In the case of this circuit, a different original circuit may yield a better optimization. In general, this problem of separate optimization and mapping is handled by using combined optimization steps and mapping steps in commercial logic optimization tools.

REFERENCES

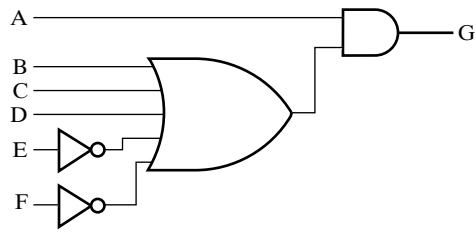
1. MANO, M. M. AND C. R. KIME. *Logic and Computer Design Fundamentals, 4th ed.* Upper Saddle River, NJ: Pearson Prentice Hall, 2008.
2. MANO, M. M. AND C. R. KIME. *Logic and Computer Design Fundamentals, 3rd ed.* Upper Saddle River, NJ: Pearson Prentice Hall, 2004.

PROBLEMS

1. Perform a low-cost (use minimum total normalized area as cost) technology mapping using cells from Table 1 for the circuit shown in Figure 3.



□ **FIGURE 3**
Logic Diagram for Problem 1.



□ **FIGURE 4**
Logic Diagram for Problem 2.

2. Repeat Problem 1 for the circuit shown in Figure 4.