Dear Professor or Other Supplement Recipient:

Cengage Learning has provided you with this product (the "Supplement") for your review and, to the extent that you adopt the associated textbook for use in connection with your course (the "Course"), you and your students who purchase the textbook may use the Supplement as described below. Cengage Learning has established these use limitations in response to concerns raised by authors, professors, and other users regarding the pedagogical problems stemming from unlimited distribution of Supplements.

Cengage Learning hereby grants you a nontransferable license to use the Supplement in connection with the Course, subject to the following conditions. The Supplement is for your personal, noncommercial use only and may not be reproduced, posted electronically or distributed except that portions of the Supplement may be provided to your students IN PRINT FORM ONLY in connection with your instruction of the Course, so long as such students are advised that they may not copy or distribute any portion of the Supplement to any third party. Test banks and other testing materials may be made available in the classroom and collected at the end of each class session, or posted electronically as described herein. Any material posted electronically must be through a password-protected site, with all copy and download functionality disabled, and accessible solely by your students who have purchased the associated textbook for the Course. You may not sell, license, auction, or otherwise redistribute the Supplement in any form. We ask that you take reasonable steps to protect the Supplement from unauthorized use, reproduction, or distribution. Your use of the Supplement indicates your acceptance of the conditions set forth in this Agreement. If you do not accept these conditions, you must return the Supplement unused within 30 days or receipt.

All rights (including without limitation, copyrights, patents, and trade secrets) in the Supplement are and will remain the sole and exclusive property of Cengage Learning and/or its licensors. The Supplement is furnished by Cengage Learning on an "as is" basis without any warranty, express or implied. This Agreement will be governed by and construed pursuant to the laws of the State of New York, without regard to such State's conflict of law rules.

Thank you for your assistance in helping to safeguard the integrity of the content contained in this Supplement. We trust you find the Supplement a useful teaching tool.
# Table of Contents

## I INTRODUCTION
1.1 Using the Text in a Lecture Course 1
1.2 Some Remarks About the Text 2
1.3 Using the Text in a Self-Paced Course 2
1.4 Use of Computer Software 4
1.5 Suggested Equipment for Laboratory Exercises 5

## II SOLUTIONS TO HOMEWORK PROBLEMS

<table>
<thead>
<tr>
<th>Unit</th>
<th>Problem Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

## III SOLUTIONS TO DESIGN, SIMULATION, AND LAB EXERCISES

<table>
<thead>
<tr>
<th>Unit</th>
<th>Problem Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>239</td>
</tr>
<tr>
<td>10</td>
<td>259</td>
</tr>
<tr>
<td>12</td>
<td>273</td>
</tr>
<tr>
<td>16</td>
<td>279</td>
</tr>
<tr>
<td>17</td>
<td>289</td>
</tr>
<tr>
<td>20</td>
<td>297</td>
</tr>
</tbody>
</table>

## IV SAMPLE UNIT TESTS

<table>
<thead>
<tr>
<th>Unit</th>
<th>Problem Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>327</td>
</tr>
</tbody>
</table>
I: INTRODUCTION

The text, Fundamentals of Logic Design, 6th edition, has been designed so that it can be used either for a standard lecture course or for a self-paced course. The text is divided into 20 study units in such a way that the average study time for each unit is about the same. The units have undergone extensive class testing in a self-paced environment and have been revised based on student feedback. The study guides and text material are sufficient to allow almost all students to achieve mastery of all of the objectives. For example, the material on Boolean algebra and algebraic simplification is 2½ units because students found this topic difficult. There is a separate unit on going from problem statements to state graphs because this topic is difficult for many students.

The textbook contains answers for all of the problems that are assigned in the study guides. This Instructor’s Manual contains complete solutions to these problems. Solutions to the remaining homework problems as well as all design and simulation exercises are also included in this manual. In the solutions section of this manual, the abbreviation FLD stands for Fundamentals of Logic Design (6th ed.).

Information on the self-paced course taught at the University of Texas using the textbook is available at www.ece.utexas.edu/projects/ee316. This website also links to an updated errata list for the text. In addition to the textbook and study guides, teaching a self-paced course requires that a set of tests be prepared for each study unit. This manual contains a sample test for each unit.

1.1 Using the Text in a Lecture Course

Even though the text was developed in a self-paced environment, the text is well suited for use in a standard lecture course. Since the format of the text differs somewhat from a conventional text, a few suggestions for using the text in a lecture course may be appropriate. Except for the inclusion of objectives and study guides, the units in the text differ very little from chapters in a standard textbook. The study guides contain very basic questions, while the problems at the end of each unit are of a more comprehensive nature. For this reason, we suggest that specific study guide questions be assigned for students to work through on their own before working out homework problems selected from those at the end of the unit. The unit tests given in Part IV of this manual provide a convenient source of additional homework assignments or a source of quiz problems. The text contains many examples that are completely worked out with detailed step-by-step explanations. Discussion of these detailed examples in lecture may not be necessary if the students study them on their own. The lecture time is probably better spent discussing general principles and applications as well as providing help with some of the more difficult topics. Since all of the units have study guides, it would be possible to assign some of the easier topics for self-study and devote the lectures to the more difficult topics.

At the University of Texas a class composed largely of Electrical Engineering and Computer Science sophomores and juniors covers 18 units (all units except 6 and 19) of the text in one semester. Units 8, 10, 12, 16, 17, and 20 contain design problems that are suitable for simulation and lab exercises. The design problems help tie together and review the material from a number of preceding units. Units 10, 17, and 20 introduce the VHDL...
hardware description language. These units may be omitted if desired since no other units depend
on them.

1.2 Some Remarks About the Text

In this text, students are taught how to use Boolean algebra effectively, in contrast with
many texts that present Boolean algebra and a few examples of its application and then leave it to
the student to figure out how to use it effectively. For example, use of the theorem \( x + yz = (x + y) \)
\((x + z)\) in factoring and multiplying out expressions is taught explicitly, and detailed guidelines are
given for algebraic simplification.

Sequential circuits are given proper emphasis, with over half of the text devoted to this
subject. The pedagogical strategy the text uses in teaching sequential circuits has proven to be very
effective. The concepts of state, next state, etc. are first introduced for individual flip-flops, next
for counters, then for sequential circuits with inputs, and finally for more abstract sequential circuit
models. The use of timing charts, a subject neglected by many texts, is taught both because it is
a practical tool widely used by logic design engineers and because it aids in the understanding of
sequential circuit behavior.

The most important and often most difficult part of sequential circuit design is formulating
the state table or graph from the problem statement, but most texts devote only a few paragraphs
to this subject because there is no algorithm. This text devotes a full unit to the subject, presents
guidelines for deriving state tables and graphs, and provides programmed exercises that help the
student learn this material. Most of the material in the text is treated in a fairly conventional
manner with the following exceptions:

1. The diagonal form of the 5-variable Karnaugh map is introduced in Unit 5. (We find that
students make fewer mistakes when using the diagonal form of 5-variable map in comparison
with the side-by-side form.) Unit 5 also presents a simple algorithm for finding all essential
prime implicants from a Karnaugh map.
2. Both the state graph approach (Unit 18) and the SM chart approach (Unit 19) for designing
sequential control circuits are presented.
3. The introduction to the VHDL hardware description language in Units 10, 17, and 20
emphasizes the relation between the VHDL code and the actual hardware.

1.3 Using the Text in a Self-Paced Course

This section introduces the personalized system of self-paced instruction (PSI) and offers
suggestions for using the text in a self-paced course. PSI (Personalized System of Instruction)
is one of the most popular and successful systems used for self-paced instruction. The essential
features of the PSI method are

(a) Students are permitted to pace themselves through the course at a rate commensurate with
their ability and available time.
(b) A student must demonstrate mastery of each study unit before going onto the next.
(c) The written word is stressed; lectures, if used, are only for motivation and not for transmission
of critical information.
(d) Use of proctors permits repeated testing, immediate scoring, and significant personal
interaction with the students.

These factors work together to motivate students toward a high level of achievement in a well-
designed PSI course

The PSI method of instruction and its implementation are described in detail in the following references:


Results of applying PSI to a first course in logic design of digital systems are described in

The instructor in charge of a self-paced course will serve as course manager in addition to his role in the classroom. For a small class, he may spend a good part of his time acting as proctor in the classroom, but as class size increases he will have to devote more of his time to supervision of course activities and less time to individual interaction with students. In his managerial role, the instructor is responsible for organizing the course, selection and training of proctors, supervision of proctors, and monitoring of student progress. The proctors play an important role in the success of a self-paced course, and therefore their selection, training, and supervision is very important. After an initial session to discuss proper ways of grading readiness tests and interacting with students, weekly proctor meetings to discuss course procedures and problems may be appropriate.

A progress chart showing the units completed by each student is very helpful in monitoring student progress through the course. The instructor may wish to have individual conferences with students who fall too far behind. The instructor needs to be available in the classroom to answer individual student questions and to assist with grading of readiness tests as needed. He should make a special point to speak with the weak or slow students and give them a word of encouragement. From time to time he may need to settle differences which arise between proctors and students.

Various strategies for organizing a PSI course are described in the Keller Plan Handbook. The procedures used for operating the self-paced digital logic course at the University of Texas are described in “Unit 0”, which is available on the web: www.ece.utexas.edu/projects/ee316. At the first class meeting, we hand out a copy of Unit 0. The students are asked to read through Unit 0 and take a short test on the course procedures. This test is immediately evaluated so that the student can complete Unit 0 before the end of the first class period. In this way, the student is exposed to the basic way the course operates and is ready to proceed immediately with Unit 1 in the textbook.

During a typical class period, some of the students will spend their time studying but most of the students will come prepared to take a unit test. At the beginning of the period, the instructor or a proctor will be available to answer student questions on an individual basis. Later in the period, most of the time will be spent evaluating unit tests. We have found that a standard 50 minute class period is not long enough for a PSI session. We usually schedule sessions of 1 ½ or 2 hours or longer depending on class size. This allows adequate time for students to have their questions answered, take a unit test, and have their tests
graded Interactive grading of the tests with the student present is an important part of the PSI system and adequate time must be allowed for this activity. If you have a large number of students and proctors, you may wish to prepare a manual for guidance of your proctors. The procedures that we use for evaluating unit tests are described in a Proctor's Manual, which can be obtained by writing to Professor Charles H. Roth.

1.4. Use of Computer Software

Three software packages are included on the CD that accompanies the textbook. The first is a logic simulator program called SimUaid, the second is a basic computer-aided logic design program called LogicAid, and the third is a VHDL Simulator called DirectVHDL. In addition, we use the Xilinx ISE software for synthesizing VHDL code and downloading to CPLD or FPGA circuit boards. The Xilinx ISE software is available at nominal cost through the Xilinx University Program (for information, go to www.xilinx.com/univ/overview.html). A “Webpack” version of the Xilinx software is also available for downloading from the Xilinx com website.

SimUaid provides an easy way for students to test their logic designs by simulating them. We first introduce SimUaid in Unit 4, where we ask the students to design a simple logic circuit such as problem 4.13 or 4.14, and simulate it. SimUaid is easy to learn, and it is highly interactive so that students can flip a simulated switch and immediately observe the result. In Unit 8, students design a multi-output combinational logic circuit using NAND and NOR gates and test its operation using SimUaid. Students can use the simulator to help them understand the operation of latches and flip-flops in Unit 11. In Unit 12, we ask them to design a counter and simulate it (one part of problem 12.10). In Unit 16, students use SimUaid to test their sequential circuit designs. They can also generate VHDL code from their SimUaid circuit, synthesize it, and download it to a circuit board for hardware testing. In Unit 18, students can use the advanced features of SimUaid to simulate a multiplier or divider controlled by a state machine.

LogicAid provides an easy way to introduce students to the use of the computer in the logic design process. It enables them to solve larger, more practical design problems than they could by hand. They can also use LogicAid to verify solutions that they have worked out by hand. Instructors can use the program for grading homework and quizzes. We first introduce LogicAid in Unit 5. The program has a Karnaugh Map Tutorial mode that is very useful in teaching students to solve Karnaugh map problems. This tutorial mode helps students learn to derive minimum solutions from a Karnaugh map by informing them at each step whether that step is correct or not. It also forces them to choose essential prime implicants first. When in the KMap tutor mode, LogicAid prints “KMT” at the top of each output page, so you can check to see if the problems were actually solved in the tutorial mode.

Students can use LogicAid to help them solve design problems in Units 8, 16, 18, 19, and other units. For designing sequential circuits, they can input a state graph, convert it to a state table, reduce the state table, make a state assignment, and derive minimized logic equations for outputs and flip-flop inputs.

The LogicAid State Table Checker is useful for Units 14 and 16, and for other units in which students construct state tables. It allows students to check their solutions without revealing the correct answers. If the solution is wrong, the program displays a short input sequence for which the student's table fails. The LogicAid folder on the CD contains encoded copies of solutions for most of the state graph problems in Fundamentals of Logic Design, 6th Ed. If you wish to create a password-protected solution file for other state table problems, enter the state table into LogicAid, syntax.
check it, and then hold down the Ctrl key while you select Save As on the file menu. The Partial Graph Checker serves as a state graph tutor that allows a student to check his work at each step while constructing a state graph. If the student makes a mistake, it provides feedback so that the student can correct his answer. The partial graph checker works with any state graph problem for which an encoded state table solution file is provided.

The DirectVHDL simulator helps students learn VHDL syntax because it provides immediate visual feedback when they make mistakes. Our students use it for simulating VHDL code in Units 10, 17, and 20. Students can simulate and debug their code at home and then bring the code into lab for synthesis and hardware testing.

1.5. Suggested Equipment for Laboratory Exercises

Many types of logic lab equipment are available that are adequate to perform the lab exercises. Since most logic design is done today using programmable logic instead of individual ICs, we now recommend use of CPLDs or FPGAs for hardware implementation of logic circuit designs. At the University of Texas, we are presently using the XILINX Spartan-3 FPGA boards, which are available from Digilent. The Spartan-3 FPGA has more than an adequate number of logic cells to implement the lab exercises in the text. The board has 8 switches, 4 pushbuttons, 8 single LEDs, and four 7-segment LEDs. Information about this board and other CPLD and FPGA boards made by Digilent can be found on their website, www.digilentinc.com. We use the board in conjunction with the Xilliinx ISE software mentioned earlier.
II. SOLUTIONS TO HOMEWORK PROBLEMS

Unit 1 Problem Solutions

1.1 (a) \[ 757.25_{10} \]

\[
\begin{array}{c|cc}
16 & 757 & 0.25 \\
16 & 47 & 16 \\
16 & 2 & r_{15} = F_{16} (4.00) \\
0 & r_2 & \\
\end{array}
\]

\[ \therefore 757.25_{10} = 2F.50_{16} = 0010 1111 0101.0100 0000_{2} \]

1.1 (b) \[ 123.17_{10} \]

\[
\begin{array}{c|cc}
16 & 123 & 0.17 \\
16 & 7 & r_{11} = 16 \\
16 & 5 & r_7 (2.72) \\
0 & r_2 & \\
\end{array}
\]

\[ \therefore 123.17_{10} = 7B.2B_{16} = 0111 1011.0010 1011_{2} \]

1.1 (c) \[ 356.89_{10} \]

\[
\begin{array}{c|cc}
16 & 356 & 0.89 \\
16 & 22 & r_4 = 16 \\
16 & 1 & r_6 (14.24) \\
0 & r_1 & \\
\end{array}
\]

\[ \therefore 356.89_{10} = 164.E3_{16} = 0001 0110 0100 1110 0011_{2} \]

1.1 (d) \[ 1063.5_{10} \]

\[
\begin{array}{c|cc}
16 & 1063 & 0.5 \\
16 & 66 & r_7 = 16 \\
16 & 4 & r_2 (8.00) \\
0 & r_4 & \\
\end{array}
\]

\[ \therefore 1063.5_{10} = 427.81_{16} = 0100 0010 0111 1000_{2} \]

1.2 (a) \[ EB1.6_{16} = 5 \times 16^2 + 9 \times 16^1 + D \times 16^0 + C \times 16^{-1} \]

\[ = 5 \times 256 + 9 \times 16 + 13 + 6/16 = 3761.375_{10} \]

\[ \therefore EB1.6_{16} = 11 23 01.01 304_{2} \]

1.2 (b) \[ 59D.C_{16} = 5 \times 16^2 + 9 \times 16^1 + D \times 16^0 + C \times 16^{-1} \]

\[ = 5 \times 256 + 9 \times 16 + 13 + 12/16 = 1437.75_{10} \]

\[ \therefore 59D.C_{16} = 11 23 01.01 304_{2} \]

1.3 \[ 3BA.25_{14} = 3 \times 14^2 + 11 \times 14^1 + 10 \times 14^0 + 2 \times 14^{-1} + 5 \times 14^{-2} \]

\[ = 588 + 154 + 10 + 0.1684 = 752.1684_{10} \]

\[
\begin{array}{c|cc}
6 & 752 & 0.1684 \\
6 & 125 & r_2 = 6 \\
6 & 20 & r_5 (1.0104) \\
6 & 3 & r_2 = 6 \\
0 & r_3 & (0.0624) \\
\end{array}
\]

\[ \begin{array}{c|cc}
0 & 624 & 0.3744 \\
0 & 246 & (2.464) \\
0 & 4 & (1.4784) \\
\end{array}
\]

\[ \therefore 3BA.25_{14} = 752.1684_{10} = 3252.1002_{6} \]

1.4 (b) \[ 1457.11_{10} \]

\[
\begin{array}{c|cc}
16 & 1457 & 0.11 \\
16 & 91 & r_1 = 16 \\
16 & 5 & r_{11} = B_{16} (1.76) \\
0 & r_5 & \\
\end{array}
\]

\[ \therefore 1457.11_{10} = 5B1.1C_{16} \]

\[ 5B1.1C_{16} = 0101 1011.0001 0001 1100_{2} = 2661.0708_{8} \]

1.4 (c) \[ 5B1.1C_{16} = 11 23 01.01 30_{2} \]

\[ \begin{array}{c|cc}
5 & 23 & 01 01 30 & \\
B & 1 & C & \\
\end{array}
\]

1.4 (d) \[ DEC.A_{16} = D \times 16^2 + E \times 16^1 + C \times 16^0 + A \times 16^{-1} \]

\[ = 3328 + 224 + 12 + 0.625 = 3564.625_{10} \]
Unit 1 Solutions

1.10 (a) \[ \begin{array}{l} 1305.375_{10} \times 1010 \quad 1111_{(Multiplication)} \\
+1010 \quad 1111_{(Add)} \\
-1010 \quad 1111_{(Sub)} \\
11001 \quad 0101 \\
\end{array} \]

\[ \therefore 1305.375_{10} + 10 = 1315.375_{10} \]

1.10 (b) \[ \begin{array}{l} 1111_{16} \times 16 \quad 16 \quad 0.33 \\
6 \quad r15 = F_{16} \\
(5.28) \quad 16 \\
(4.48) \end{array} \]

\[ \therefore 111.33_{10} = 6F.54_{16} \]

1.10 (c) \[ \begin{array}{l} 301.12_{10} \times 18 \quad 16 | 301 \\
16 | 18 \quad 0.12 \\
5 \quad r1 \quad (1.92) \\
1 \quad r2 \quad (14.72) \\
\end{array} \]

\[ \therefore 301.12_{10} = 12D.1E_{16} \]

1.10 (d) \[ \begin{array}{l} 1644.875_{10} \times 102 \quad 16 | 1644 \\
16 | 102 \quad 0.875 \\
6 \quad r6 \quad (14.000) \\
\end{array} \]

\[ \therefore 1644.875_{10} = 66C.E00_{16} \]

1.11 (a) \[ \begin{array}{l} 100.101_{2} = 5724.5_{8} \\
= 5 \times 8^3 + 7 \times 8^2 + 2 \times 8 + 4 + 5 \times 8^{-1} \\
= 5 \times 512 + 7 \times 64 + 2 \times 8 + 4 + 5/8 \\
= 3028.625_{10} \\
\end{array} \]

1.12 (a) \[ \begin{array}{l} 375.54_{8} = 3 \times 64 + 7 \times 8 + 5 + 5/8 + 4/64 \\
= 253.6875_{10} \\
3 | 253 \quad 0.69 \\
3 | 84 \quad r1 \\
3 | 28 \quad r0 \quad (2.07) \\
3 | 9 \quad r1 \\
3 | 3 \quad r0 \quad (0.21) \\
3 | 1 \quad r0 \\
0 \quad r1 \quad (0.63) \\
\end{array} \]

\[ \therefore 375.54_{8} = 100101.2001_{5} \]
1.12 (c) $A52.A4_{10} = 10 \times 121 + 5 \times 11 + 2 + 10/11 + 4/121$
\[= 1267.94 \text{,}_10\]

\[9 \mid 1267 \quad 0.94\]
\[9 \mid 140 \quad r7 \quad 9\]
\[9 \mid 15 \quad r5 \quad (8.46)\]
\[9 \mid 1 \quad r6 \quad 9\]
\[0 \quad r1 \quad (4.14)\]

∴ $A52.A4_{10} = 1267.94_{10} = 1657.84_{27\ldots}^g$

1.14 (a), (c) $16 \mid 97 \quad .7$
(b), (e) $16 \mid 16 \quad r1 \quad 16$
\[0 \quad r6 \quad (1\text{1.2})\]
\[- \quad 16 \quad (3.2)\]

∴ $97.7_{10} = 61.B333\ldots_{16}$
(a) $61.B333\ldots_{16}$
\[= 110 0001.1011 0011 0011 0011\ldots_2\]
(b) $110 001.101 100 100 110 111\ldots_2$
\[= 141.5 4631 4631\ldots_8\]

1.14 (e) $5 \mid 97 \quad .7$
\[5 \mid 19 \quad r2 \quad 5\]
\[5 \mid 3 \quad r4 \quad (3.5)\]
\[0 \quad r3 \quad (2.5)\]

∴ $97.7_{10} = 342.322\ldots_5$

1.15 \[1110212.202110_2 = 1425.673\_9\]

1.16 (a) $2983.63/64_{10} = \frac{2983}{64} = 0.984$
\[8 \mid 2983 \quad 0.984\]
\[8 \mid 372 \quad r7 \quad 8\]
\[8 \mid 46 \quad r4 \quad (7.872)\]
\[9 \mid 5 \quad r6 \quad 8\]
\[0 \quad r5 \quad (6.976)\]

∴ $2983.63/64_{10} = 5647.76_{8}$ (or 5647.77_{8})
\[= 101 110 100 111.111 110_2\]
\[= (or 101 110 100 111.111 111.2)_{2}\]

1.16 (b) $93.70_{10} = 8 \mid 93 \quad 0.70$
\[8 \mid 11 \quad r5 \quad 8\]
\[8 \mid 1 \quad r3 \quad (5.60)\]
\[0 \quad r1 \quad 8\]

∴ $93.70_{10} = 135.54_{8} = 001 011 101.101 100\ldots_2$
Unit 1 Solutions

1.16 (c)  $1900 \div 32_{10}$

```
8 | 1900  0.969
  8  273 r4  8
  8  29 r5  (7.752
  9  3 r5  8
  0  r3  (6.016

\[1900 \div 32 = 52.8125\]
```

1.16 (d)  $109.30_{10}$

```
8 | 109  0.30
  8  13 r5  8
  8  1 r5  (2.40
  0  r1  8

\[109.30 \div 32 = 3.409375\]
```

\[\therefore 1900 \div 32_{10} = 3554.768_{10}\]

\[= 011 101 101 100.111 110_{2}\]

\[\therefore 109.30_{10} = 155.23_{8}\]

\[= 001 101 101.010 011_{2}\]

1.17 (a)

```
1111 (Add)  1111 (Subtract)
  1001
  1001
  11000
  0110

1111 (Multiply)
  1001
  1111
  0000
  01111
  0000
  001111
  1111
  10000111
```

1.17 (b)

```
1101001 (Add)  1101001 (Sub)
  110110
  110110
  10011111
  10011111

1101001 (Multiply)
  110110
  1101001
  1101001
  10011111
  10011111
  0000000
  1101001
  100100000110
  1101001
  1011000100110
```

1.17(c)

```
110010 (Add)  110010 (Sub)
  11101
  11101
  1001111
  10101

110010 (Multiply)
  11101
  110010
  000000
  0110010
  110010
  11111010
  10010001010
  110010
  10110101010
```

1.18

(a)  10100100
    01110011
(b)  10010011
    01011001
(c)  11110011
    10011110
    01010101

1.19(a)

```
101110 Quotient
  101
  101
  101
  101
  101
  11

11 Remainder
```

1.19(b)

```
11011 Quotient
  11110000001
  1110
  10100
  1110
  1110
  110

11 Remainder
```

\[\therefore 1900 \div 32_{10} = 3554.768_{10}\]

\[= 011 101 101 100.111 110_{2}\]

\[\therefore 109.30_{10} = 155.23_{8}\]

\[= 001 101 101.010 011_{2}\]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
1100 Quotient
\[ \begin{array}{c|c}
101 & 1110010 \\
1001 & 1010 \\
101 & 110 \\
110 & 110 \\
101 & 110 \\
101 & 10 \\
110 & 10 \\
\end{array} \]
10 Remainder

100011 Quotient
\[ \begin{array}{c|c}
1011 & 11000011 \\
1011 & 1011 \\
1011 & 110 \\
1011 & 1011 \\
1011 & 1011 \\
1011 & 1011 \\
110 & 1011 \\
\end{array} \]
10 Remainder

1.21 (a) \(4 + 3 = 10\) in base 7, i.e., the sum digit is 0 with a carry of 1 to the next column. \(1 + 5 + 4 = 10\) in base 7. 1 + 6 + 0 = 10 in base 7. This overflows since the correct sum is 10007.
(b) \(4 + 3 + 3 + 3 = 13\) in base 10 and 23 in base 5. Try base 10. \(1 + 2 + 4 + 1 + 3 = 11\) in base 10 so base 10 does not produce a sum digit of 2. Try base 5. \(2 + 2 + 4 + 1 + 3 = 22\) in base 5 so base 5 works.
(c) \(4 + 3 + 3 + 3 = 31\) in base 4, 21 in base 6, and 11 in base 12. Try base 12. \(1 + 2 + 4 + 1 + 3 = B\) in base 12 so base 12 does not work. Try base 4. \(3 + 2 + 4 + 1 + 3 = 31\) in base 4 so base 4 does not work. Try base 6. \(2 + 2 + 4 + 1 + 3 = 20\) so base 6 is correct.

1.24 (a) Expand the base \(b\) number into a power series
\[ N = d_{3k-1}b^{3k-1} + d_{3k-2}b^{3k-2} + d_{3k-3}b^{3k-3} + \ldots + d_2b^2 + d_1b^1 + d_0b^0 \]
where each \(d_i\) has a value from 0 to \((b-1)\).

1.24 (b) Expand the base \(b^3\) number into a power series
\[ N = d_k(b^3)^k + d_{k-1}(b^3)^{k-1} + \ldots + d_1(b^3)^1 + d_0(b^3)^0 + d_{3m-2}b^2 + d_{3m-1}b^1 + d_{3m}b^0 \]
where each \(d_i\) has a value from 0 to \((b^3-1)\). Consequently, \(d_i\) can be represented as a base \(b\) number in the form
\[ e_{3i-1}b^2 + e_{3i-2}b^1 + e_{3i-3}b^0 \]
where each \(e_i\) has a value from 0 to \((b-1)\).

1.20(a)
\[ \begin{array}{c|c}
1011 & 10001101 \\
1011 & 1011 \\
1010 & 1011 \\
1010 & 1011 \\
1010 & 1011 \\
110 & 1010 \\
\end{array} \]
11 Remainder

1.20(c)
\[ \begin{array}{c|c}
1011 & 1110100 \\
1010 & 11010 \\
1010 & 10000 \\
1010 & 110 \\
110 & 110 \\
\end{array} \]
10 Remainder

1.22 If the binary number has \(n\) bits (to the right of the radix point), then its precision is \((1/2^{n+1})\). So to have the same precision, \(n\) must satisfy
\[ (1/2^{n+1}) < (1/2)(1/104) \]
so \(n > 4/(\log 2) = 13.28\) so \(n\) must be 14.

1.23 \(0.363636\ldots = (36/102)(1 + 1/10^2 + 1/10^4 + 1/10^6 + \ldots) = (36/102)(1/(1 - 1/10^2)) = (36/102)(10^2/99) = 36/99 = 4/11\)
8(4/11) = 2 + 10/11
8(10/11) = 7 + 3/11
8(3/11) = 2 + 2/11
8(2/11) = 1 + 5/11
8(5/11) = 3 + 7/11
8(7/11) = 5 + 1/11
8(1/11) = 0 + 8/11
8(8/11) = 5 + 9/11
8(9/11) = 6 + 6/11
8(6/11) = 4 + 4/11
8(4/11) = 2 + 10/11
Repeats: \(0.27213505642\ldots\)

1.20(b)
\[ \begin{array}{c|c}
1011 & 110000011 \\
1011 & 1011 \\
1010 & 1011 \\
1010 & 1011 \\
1010 & 1011 \\
110 & 110 \\
\end{array} \]
11 Remainder

1.20(c)
\[ \begin{array}{c|c}
1011 & 1110100 \\
1010 & 10010 \\
1010 & 10000 \\
1010 & 1010 \\
1010 & 110 \\
\end{array} \]
10 Remainder

1.20(b)
\[ \begin{array}{c|c}
1101 & 1001001 \\
1100 & 11010 \\
1101 & 10000 \\
1101 & 1010 \\
1101 & 110 \\
\end{array} \]
10 Remainder

1.20(a)
\[ \begin{array}{c|c}
1011 & 100001101 \\
1011 & 1011 \\
1010 & 1011 \\
1010 & 1011 \\
1010 & 1011 \\
110 & 110 \\
\end{array} \]
11 Remainder

1.20(c)
\[ \begin{array}{c|c}
1011 & 1001010 \\
1010 & 10000 \\
1010 & 1010 \\
110 & 110 \\
\end{array} \]
10 Remainder

1.20(a)
\[ \begin{array}{c|c}
1011 & 100110101 \\
1011 & 1011 \\
1010 & 1011 \\
1010 & 1011 \\
1010 & 1011 \\
110 & 110 \\
\end{array} \]
11 Remainder

1.24 (b) Expand the base \(b^3\) number into a power series
\[ N = d_k(b^3)^k + d_{k-1}(b^3)^{k-1} + \ldots + d_1(b^3)^1 + d_0(b^3)^0 + d_{3m-2}b^2 + d_{3m-1}b^1 + d_{3m}b^0 \]
where each \(d_i\) has a value from 0 to \((b^3-1)\). Consequently, \(d_i\) can be represented as a base \(b\) number in the form
\[ e_{3i-1}b^2 + e_{3i-2}b^1 + e_{3i-3}b^0 \]
where each \(e_i\) has a value from 0 to \((b-1)\). Substituting these expressions for the \(d_i\) produces a power series expansion for a base \(b\) number.
Unit 1 Solutions

1.25

<table>
<thead>
<tr>
<th>4 3 2 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 0 0 0</td>
</tr>
<tr>
<td>2 0 0 1</td>
</tr>
<tr>
<td>3 0 1 0</td>
</tr>
<tr>
<td>4 1 0 0</td>
</tr>
<tr>
<td>5 1 0 1</td>
</tr>
<tr>
<td>6 1 1 0</td>
</tr>
<tr>
<td>7 1 1 0</td>
</tr>
<tr>
<td>8 1 1 0</td>
</tr>
<tr>
<td>9 1 1 1</td>
</tr>
</tbody>
</table>

1.26

5-3-1-1 is possible, but 6-4-1-1 is not, because there is no way to represent 3 or 9.

Alternate Solutions:

<table>
<thead>
<tr>
<th>5 3 1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
</tr>
<tr>
<td>2 0 1 0</td>
</tr>
<tr>
<td>3 0 1 1</td>
</tr>
<tr>
<td>4 1 0 1</td>
</tr>
<tr>
<td>5 1 1 0</td>
</tr>
<tr>
<td>6 1 1 1</td>
</tr>
<tr>
<td>7 1 1 1</td>
</tr>
<tr>
<td>8 1 1 1</td>
</tr>
<tr>
<td>9 1 1 1</td>
</tr>
</tbody>
</table>

9154 = 1110 0001 1010 1000

1.27

5-4-1-1 is not possible, because there is no way to represent 3 or 8. 6-3-2-1 is possible:

Alternate Solutions:

<table>
<thead>
<tr>
<th>6 3 2 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
</tr>
<tr>
<td>2 0 1 0</td>
</tr>
<tr>
<td>3 0 1 1</td>
</tr>
<tr>
<td>4 1 0 1</td>
</tr>
<tr>
<td>5 1 1 0</td>
</tr>
<tr>
<td>6 1 1 1</td>
</tr>
<tr>
<td>7 1 1 1</td>
</tr>
<tr>
<td>8 1 1 1</td>
</tr>
<tr>
<td>9 1 1 1</td>
</tr>
</tbody>
</table>

1.28

Alternate Solutions:

<table>
<thead>
<tr>
<th>6 2 2 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
</tr>
<tr>
<td>2 0 1 0</td>
</tr>
<tr>
<td>3 0 1 1</td>
</tr>
<tr>
<td>4 1 0 1</td>
</tr>
<tr>
<td>5 1 1 0</td>
</tr>
<tr>
<td>6 1 1 1</td>
</tr>
<tr>
<td>7 1 1 1</td>
</tr>
<tr>
<td>8 1 1 1</td>
</tr>
<tr>
<td>9 1 1 1</td>
</tr>
</tbody>
</table>

1.29

Alternate Solutions:

<table>
<thead>
<tr>
<th>5 2 2 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
</tr>
<tr>
<td>2 0 1 0</td>
</tr>
<tr>
<td>3 0 1 1</td>
</tr>
<tr>
<td>4 1 0 1</td>
</tr>
<tr>
<td>5 1 1 0</td>
</tr>
<tr>
<td>6 1 1 1</td>
</tr>
<tr>
<td>7 1 1 1</td>
</tr>
<tr>
<td>8 1 1 1</td>
</tr>
<tr>
<td>9 1 1 1</td>
</tr>
</tbody>
</table>

1100 0011 = 83

1.30

Alternate Solutions:

<table>
<thead>
<tr>
<th>7 3 2 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
</tr>
<tr>
<td>2 0 1 0</td>
</tr>
<tr>
<td>3 0 1 1</td>
</tr>
<tr>
<td>4 1 0 1</td>
</tr>
<tr>
<td>5 1 1 0</td>
</tr>
<tr>
<td>6 1 1 1</td>
</tr>
<tr>
<td>7 1 1 1</td>
</tr>
<tr>
<td>8 1 1 1</td>
</tr>
<tr>
<td>9 1 1 1</td>
</tr>
</tbody>
</table>

1110 0110 = 94

1.31

(a)

<table>
<thead>
<tr>
<th>8 4-2-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 0 1 1</td>
</tr>
<tr>
<td>2 0 1 0</td>
</tr>
<tr>
<td>3 0 1 1</td>
</tr>
<tr>
<td>4 1 0 0</td>
</tr>
<tr>
<td>5 1 0 1</td>
</tr>
<tr>
<td>6 1 1 0</td>
</tr>
<tr>
<td>7 1 0 1</td>
</tr>
<tr>
<td>8 1 0 0</td>
</tr>
<tr>
<td>9 1 1 1</td>
</tr>
</tbody>
</table>

(b) The 9’s complement of a decimal number represented with this weighted code can be obtained by replacing 0’s with 1’s and 1’s with 0’s (bit-by-bit complement).

B4A9 = 1101 0101 1100 1010
Alt. = " " 1011 "

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
1.32 (a) 222.22\textsubscript{10}

<table>
<thead>
<tr>
<th>16</th>
<th>222</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>0</td>
<td>r13</td>
</tr>
</tbody>
</table>

\[ \begin{array}{c|c}
1.22 & 0.22 \\
\hline
1.16 & (3.52) \\
0 & (8.32)
\end{array} \]

\[ \therefore 222.22 = \text{DE.38}_{16} \]

\[ = 1000100 \quad 1000101 \quad 0100110 \quad 0110011 \quad 0111000 \]

D E . 3 8

1.32 (b) 183.81\textsubscript{10}

<table>
<thead>
<tr>
<th>16</th>
<th>183</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>r7</td>
</tr>
</tbody>
</table>

\[ \begin{array}{c|c}
1.81 & 0.16 \\
\hline
1.16 & (12.96) \\
0 & (15.36)
\end{array} \]

\[ \therefore 183.81 = \text{B7.CF}_{16} \]

\[ = 1000010 \quad 0110111 \quad 0101110 \quad 1000111 \quad 1000110 \]

B 7 . C F

1.33 (a) In 2’s complement: \(-10\) + \(-11\)

| 110110 |
| 110101 |
| (1)10101 (-21) |

In 1’s complement: \(-10\) + \(-11\)

| 110101 |
| 110100 |
| (1)10101 (-21) |

\[ \begin{array}{c}
101010 (-21)
\end{array} \]

1.33 (b) In 2’s complement: \(-10\) + \(-6\)

| 110110 |
| 110101 |
| (1)101000 (-16) |

In 1’s complement: \(-10\) + \(-6\)

| 110101 |
| 110000 |
| (1)101110 |

\[ \begin{array}{c}
101111 (-16)
\end{array} \]

1.33 (c) In 2’s complement: \(-8\) + \(-11\)

| 111000 |
| 110101 |
| (1)101101 (-19) |

In 1’s complement: \(-8\) + \(-11\)

| 110101 |
| 110100 |
| (1)10101 (-19) |

\[ \begin{array}{c}
101100 (-19)
\end{array} \]

1.33 (d) In 2’s complement: \(-11\) + \(-4\)

| 110101 |
| 111100 |
| (1)110001 (-15) |

In 1’s complement: \(-11\) + \(-4\)

| 110101 |
| 110111 |
| (1)101111 |

\[ \begin{array}{c}
110000 (-15)
\end{array} \]

1.34 (a) In 2’s complement: \(11 + 9\)

| 11010 |
| 11010 |

\[ \begin{array}{c}
01001 - 11010
\end{array} \]

In 1’s complement: \(11 + 9\)

| 11010 |
| 11010 |

\[ \begin{array}{c}
01001 - 11010
\end{array} \]

1.34 (b) In 2’s complement: \(11001 - 11010\)

| 11010 |
| 11010 |

\[ \begin{array}{c}
00111
\end{array} \]

In 1’s complement: \(11001 - 11010\)

| 11010 |
| 11010 |

\[ \begin{array}{c}
01111
\end{array} \]

1.34 (c) In 2’s complement: \(10110\)

| 10110 |

\[ \begin{array}{c}
overflow
\end{array} \]

In 1’s complement: \(10110\)

| 10110 |

\[ \begin{array}{c}
overflow
\end{array} \]

1.34 (d) In 2’s complement: \(11011 + 11001\)

| 11011 |
| 11000 |

\[ \begin{array}{c}
10100
\end{array} \]

In 1’s complement: \(11011 + 11001\)

| 11011 |
| 11000 |

\[ \begin{array}{c}
00111
\end{array} \]

1.34 (e) In 2’s complement: \(01011 + 01011\)

| 01011 |

\[ \begin{array}{c}
overflow
\end{array} \]

In 1’s complement: \(01011 + 01011\)

| 01011 |

\[ \begin{array}{c}
overflow
\end{array} \]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Unit 1 Solutions

1.35 (a) In 2’s complement | In 1’s complement
---|---
11010 | 11010
+ 01100 | + 01011
(1)00110 | (1)00101

1.35 (b) In 2’s complement | In 1’s complement
---|---
01011 | 01011
+ 01000 | + 00111
10011 | 10010

1.35 (c) In 2’s complement | In 1’s complement
---|---
10001 | 10001
+ 10110 | + 10101
(1)00111 | (1)00110
overflow | overflow

1.35 (d) In 2’s complement | In 1’s complement
---|---
10101 | 10101
+ 00110 | + 00101
11010 | 11010

1.36 (a) add | subt
-------------|-------------
101010 | 101010
+ 011101 | - 011101
(1)000111 | 001101
001000 | overflow

1.37 (a) complement
---
i) 00000000 (0) 11111111 (-0)
i) 11111110 (-1) 00000001 (1)
i) 00110011 (51) 11001100 (-51)
i) 10000000 (-127) 01111111 (127)

(b) add | subt
-------------|-------------
101010 | 101010
+ 011101 | - 011101
(1)000111 | 001101
overflow

(i) 00000000 (0) 00000000 (0)
ii) 11111110 (-2) 00000010 (2)
iii) 00110011 (51) 11001101 (-51)
iv) 10000000 (-128) 10000000 (-128)
Unit 2 Problem Solutions

2.1  
*See FLD p. 693 for solution.*

2.2 (a)  
In both cases, if X = 0, the transmission is 0, and if X = 1, the transmission is 1.

2.3  
*Answer is in FLD p. 693*

2.4 (a)  
\[ F = [(A \cdot 1) + (A \cdot 1)] + E + BCD = A + E + BCD \]

2.4 (b)  
\[ Y = (AB' + (AB + B)) B + A = (AB' + B) B + A = (A + B) B + A = AB + B + A = A + B \]

2.5 (a)  
\[ (A + B' + C') (A' + C' + D) (B' + D') \]
\[ = (A' + C' + BD) (B' + D') \]
\[ = AB + C'B + B'BD + A'D' + C'D' + BDD' = A'B' + A'D' + C'B + C'D' \]

2.5 (b)  
\[ \text{By Th. 8D with } X = A' + C' \]
\[ = A'B' + B'C'B' + B'BD + A'D' + C'D' + BDD' \]
\[ = A'B' + A'D' + C'B + C'D' \]

2.5 (c)  
\[ A'B'C + E \]
\[ = A'BC + E \]

2.6 (a)  
\[ AB + C'D' = (AB + C') (AB + D') \]
\[ = (A + C') (B + D') \]

2.6 (b)  
\[ WX + WYX + ZYX = X(W + WY + ZY) \]
\[ = X(W + Z) (W + Y) \]

2.7 (a)  
\[ (A + B + C + D) (A + B + C + E) (A + B + C + F) \]
\[ = A + B + C + D + E + F \]

Apply second distributive law (Th. 8D) twice

2.8 (a)  
\[ [(AB)' + C'D] = AB(C'D') = AB(C + D') \]
\[ = ABC + ABD' \]

2.8 (b)  
\[ [(A + B (C' + D))]' = A'(B'C + D) \]
\[ = A'(B + C') + A'B \]

2.8 (c)  
\[ [(A + B) C] = (A + B) C' = (A'B + C') \]
\[ = AB'C' \]

2.9 (a)  
\[ F = [(A + B' + (A + (A + B)))] (A + (A + B))' \]
\[ = (A + (A + B))' \]
\[ \text{By Th. 10D with } X = A + (A + B) = A'B' \]

2.9 (b)  
\[ G = [(R + S + T) PT(R + S)] T' \]
\[ = (R + S + T) PT(R + S) + T' \]
\[ = T' + (R'S'T) PS(T'S'T') \]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Unit 2 Solutions

2.10 (a) 

2.10 (b) 

2.10 (c) 

2.10 (d) 

2.10 (e) 

2.10 (f) 

2.11 (a) 

2.11 (c) 

2.11 (e) 

2.12 (a) 

2.12 (c) 

2.12 (e) 

2.13 (a) 

2.13 (c) 

2.14 (a) 

2.15 (a) 

2.16 (a) 

2.17 (a) 

2.17 (c) 

2.18 (a) product term, sum-of-products, product-of-sums)
2.18 (b) sum-of-products

2.18 (d) sum term, sum-of-products, product-of-sums

2.19

2.20 (a) \( F = D[(A' + B')C + AC'] \)

2.20 (b) \( F = D[(A' + B')C + AC'] \)

2.20 (c) \( F = D[(A' + B')C + AC'] \)

2.20 (e) product-of-sums

2.21

2.22 (a) \( A'B' + A'CD + A'DE' \)

2.22 (b) \( H'I' + JK \)

2.22 (c) \( A'BC + A'B'C + CD' \)

2.22 (d) \( A'B' + (CD' + E) = A'B' + (C + E)(D' + E) \)

2.22 (e) \( A'B'C + B'CD' + EF' = A'B'C + B'CD' + EF' \)

2.22 (f) \( WXY + W'X' + WY' = X(WY + W') + WY' \)

2.23 (a) \( W + U'YV = (W + U')(W + Y)(W + V) \)

2.23 (b) \( TW + UY' + V = (T + U + Z)(T + Y' + V)(W + U + V)(W + Y' + V) \)

2.23 (c) \( ABC + ADE' + ABF' = A(BC + DE' + BF') \)

2.23 (d) \( ABC + ADE' + ABF' = A(BC + DE' + BF') \)

2.23 (e) none apply

2.23 (f) none apply
Unit 2 Solutions

2.24 (a) \[ (XY)' + (X' + Y)Z = X'Y + (X' + Y)Z \]
\[ = X'Y + Z \text{ By Th. 11D with } Y = (X' + Y) \]

2.24 (c) \[ (A' + B)' + (A'B'C)' + C'D)' \]
\[ = (A' + B)'A'B'C(C + D)' = A'B'C \]

2.25 (a) \[ F(P, Q, R, S)' = [(R' + PQ)S]' = R(P' + Q') + S' \]
\[ = RP' + RQ' + S' \]

2.25 (c) \[ F(A, B, C, D)' = [A' + B' + ACD]' \]
\[ = [A' + B' + CD]' = AB(C' + D') \]

2.26 (a) \[ F = [(A' + B)B]'C + B = [A' + B + B']C + B = C + B \]

2.26 (c) \[ H = [WX(Y' + Z')]' = W + X + YZ \]

2.28 (a) \[ F = ABC + A'BC + AB'C + ABC' \]
\[ = BC + AB'C + ABC' \text{ (By Th. 9)} \]
\[ = C(B + AB') + AB'C = C(A + B) + ABC' \]
\[ = AC + BC + ABC' = AC + B(C + AC') \]
\[ = AC + B(A + C) = AC + AB + BC \]

Alternate solutions:
\[ F = AB + C(A + B) \]
\[ F = AC + B(A + C) \]

2.24 (b) \[ (X + (Y(Z + W))')' = XY(Z + W) = X'Y'Z'W' \]

2.24 (d) \[ (A + B)CD + (A + B)' = CD + (A + B)' \]
\[ = CD + A'B' \]

2.25 (b) \[ F(W, X, Y, Z)' = [X + YZ(W + X')]' \]
\[ = [X + X'YZ + WYZ]' = X'Y' + X'Z' \]

2.26 (b) \[ G = [(AB)'(B + C)]'C = (AB + B'C)C = ABC \]

2.27 \[ F = (V + X + W)(V + X + Y)(V + Z) \]
\[ = (V + X + W)(V + X + Y)(V + Z) = V + Z (X + WY) \]
\[ \text{By Th. 8D with } X = V \]

2.28 (b) Beginning with the answer to (a):
\[ F = A(B + C) + BC \]

Alternate solutions:
\[ F = AB + C(A + B) \]
\[ F = AC + B(A + C) \]
2.29 (a) \[ X \quad Y \quad Z \quad X+Y \quad X'+Z \quad (X+Y)(X'+Z) \quad XZ \quad XY \quad XZ+XY \]
\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

2.29 (b) \[ X \quad Y \quad Z \quad X+Y \quad Y+Z \quad X'+Z \quad (X+Y)(Y+Z)(X'+Z) \]
\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

2.29 (c) \[ X \quad Y \quad Z \quad X' \quad Y \quad Z' \quad X'Y+Z' \quad X'Z+Y' \quad X'Y+Z' \quad X'Z+Y' \]
\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

2.29 (d) \[ A \quad B \quad C \quad A+C \quad AB+C' \quad (A+C)(AB+C') \quad AB \quad AC' \quad AB+AC' \]
\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

2.29 (e) \[ W \quad X \quad Y \quad Z \quad W' \quad X' \quad Y \quad Z' \quad W'X'Y \quad W'X'Z \quad W'X'Y'Z \quad W'X'YZ \]
\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

2.30 \[ F = (X+Y)Z + X'YZ' \quad (\text{from the circuit}) \]
\[ = (X+Y)Z + X'YZ' \quad \text{(distributive law)} \]
\[ = (X+Y)(X+Y')Z + XYZ + ZY + ZY' \quad \text{(distributive law)} \]
\[ = (X+Y)(X+Y')Z + XYZ + ZY + ZY' \quad \text{(complementation laws)} \]
\[ = (X+Y)(X+Y')Z + XYZ + ZY + ZY' \quad \text{(0 and 1 operations)} \]
\[ = (X+Y)(X+Y')Z + XYZ + ZY + ZY' \quad \text{(0 and 1 operations)} \]

\[ G = (X+Y') + Z' \quad (\text{from the circuit}) \]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Unit 3 Problem Solutions

3.6 (a) \((W' + X + Z') (W'' + Y') (W'' + X + Z') (W' + Y + Z)\)
\[= (W' + X') (W'' + Y') (W'' + X + Z') (W' + Y + Z)\]
\[= (W' + X') [W'' + Y'(X + Z')] (W' + Y + Z)\]
\[= [W' + X' (Y + Z)] [W'' + Y(X + Z')] = WY'X + WY'Z + WX'Y \quad \text{(Using } (X + Y) (X' + Z) = X'Y + XZ \text{ with } X = W\)
\[= WYX' + WYZ' + W'X'Y + W'X'Z\]

3.6 (b) \((A + B + C + D) (A' + B' + C + D') (A' + D) (B + C + D)\)
\[= (B + C + D) (A' + C') (A + D) = (B + C + D) (A'D + AC) \quad \text{(Using } (X + Y) (X' + Z) = X'Y + XZ \text{ with } X = A\)
\[= A'D' + A'DC + A'D + AC + AC = A'D + AC\]

3.7 (a) \(BCD + C'D' + B'C'D + CD\)
\[= C'D' + B'C'D = (C' + D) [C + (D' + B'D)] \quad \text{(Using } (X + Y) (X' + Z) = X'Y + XZ \text{ with } X = C\)
\[= (C' + D) [C + (D' + B') (D' + D)] = (C' + D) (C + D' + B')\]

3.7 (b) \(A'C'D' + AB'D + A'CD + BD\)
\[= D' (A'C' + AB) + D (A'C + B')\]
\[= D' [(A' + B) (A' + C')] + D [(B' + A') (B' + C')] \quad \text{(Using } XY + X'Z = (X' + Y) (X + Z) \text{ twice inside the brackets})
\[= [D + (A' + B) (A + C')] [D' + (B' + A') (B' + C')] \quad \text{(Using } XY + X'Z = (X' + Y) (X + Z) \text{ with } X = D\)
\[= (D + A' + B) (D + A + C') (D' + B' + A') (D' + B' + C) \quad \text{(Using the Distributive Law)}\]

3.8 \(F = AB \oplus [(A = D) + D] = AB \oplus (A' + D' + D) = AB \oplus (A'D' + D) = AB \oplus (A' + D)
\[= (AB)' (A' + D) + AB (A' + D) = (A + D) (A' + D) + AB (AD')\]
\[= A' + B'D + ABD' \quad \text{(Using } (X + Y) (X + Z) = X + YZ \text{ with } X = A\)
\[= A' + BD' + B'D \quad \text{(Using } X + Y + Y = X + Y\)

3.9 \(A \oplus BC = (A \oplus B) (A \oplus C)\) is not a valid distributive law. PROOF: Let \(A = 1, B = 1, C = 0.\)
LHS: \(A \oplus BC = 1 \oplus 1 \cdot 0 = 1 \oplus 0 = 1.\)
RHS: \((A \oplus B) (A \oplus C) = (1 \oplus 1) (1 \oplus 0) = 0 \cdot 1 = 0.\)

3.10 (a) \((X + W) (Y' + Z) + XW'\)
\[= (X + W) (YZ' + Y'Z) + XW'\]
\[= XYZ' + YXZ' + WYZ' + WY'Z + XW'\]
Using Consensus Theorem
\[WYZ' + WY'Z + XW'\]

3.10 (b) \((A \oplus BC) + BD + ACD = A'BC + A (BC') + BD + ACD\)
\[= A'BC + A (B' + C') + BD + ACD\]
\[= A'BC + AB' + AC' + BD + ACD\]
(Add consensus term AD, eliminate ACD)
\[= A'BC + AB' + AC' + BD\]
(Remove consensus term AD)

3.10 (c) \((A' + C' + D') (A' + B + C) (A + B + D) (A + C + D)\)
\[= (A' + C' + D') (B + C + D) (A' + B + C') (A + B + D) (A + C + D)\]
Add consensus term
\[= (A' + B + C') (A + B + D)\]
\[= (A' + C' + D') (B + C' + D) (A + C + D)\]
Removing consensus terms
3.11 \[
(A + B' + C + E') (A + B' + D' + E) (B' + C' + D' + E') = [A + B' + (C + E') (D' + E)] (B' + C' + D' + E') = (A + B' + D'E + CE) (B' + C' + D' + E') = B' + (A + D'E + CE) (C' + D' + E')
\]
\[
CD' \text{ (Add consensus term)} = B' + AC' + AD' + AE' + CD' + D'E
\]

3.12 \[
A'CD'E + A'B'D' + ABCE + ABD = A'B'D' + ABD + BCD'E
\]

Proof: LHS: \[
A'CD'E + BCD'E + A'B'D'
\]

Add consensus term to left-hand side and use it to eliminate two consensus terms.

This yields the right-hand side.

\[
\therefore \text{ LHS = RHS}
\]
Unit 3 Solutions

3.15 (d) \( (K + L + M) (K' + L' + N') (K' + L' + M') (K + L + N) = (K + L + MN) (K' + L' + M.N') \)
\( = K (L' + M.N) + K'(L + MN) \) (Th. 14 with \( X = K \) = \( KL' + KM'N' + KL + K'MN' \))

3.15 (e) \( (K + L + M) (K + M + N) (K' + L' + M') (K' + M' + N') = (K + M + LN) (K' + M' + L'N') \)
\( = (M' + L'N') + (M + LN) + K'M + K'L' \)
Alt. soln's: \( K'M' + K'M + L'M'N' + LN MN \) (or) \( KM' + K'M + KL'N' + L'M'N' \) (or) \( KM' + K'M + KL'N' + LM'N' \)

3.16 (a) \( (KL + M) + MN' = (KL) + M + KLM' + MN' = (K' + L')M + KLM' + MN' = M(K' + L') + M(KL + N') \)
\( = (M' + K' + L') (M + N' + KL) = (M' + K' + L') (M + N' + K) (M + N' + L) \)

3.16 (b) \( M(K + N) + MN + K'N = M[KN' + KN] + MN + K'N = K'MN' + K'MN + MN + K'N \)
\( = K' + MN' + N(M + K' + M') = KM'N' + N = N + KM' = (K' + N)(M' + N) \)

3.17 (a) \( x = 0 = x(0) + x(0)' = x' \)
(b) \( x = I = x(I) + x(I)' = x \)
(c) \( x = x = x(x) + x'(x)' = x + x' = 1 \)
(d) \( x = x' = x(x') + x'(x)' = 0 \)
(e) \( x = y = xy + x'y = yx + y'x = y = x \)
(f) \( (x \oplus y) \oplus z = (xy + x'y) \oplus z = (xy + x'y)z + (xy' + x'y')z' = xyz + x'y'z + xy'z' + x'y'z' \)
\( = x(x' + y'z') + x'(yz + y') = x(yz + y') + x'(yz + y') = x = (y \oplus z) \)
(g) \( (x \oplus y)' = (xy + x'y')' = (x' + y')(x + y) = x'y + xy' = x + y' = x \)

3.18 (a) \( x \oplus 0 = x(0) + x(0)' = x \)
(b) \( x \oplus I = x(I) + x(I)' = x' \)
(c) \( x \oplus x = x(x) + x'(x)' = 0 \)
(d) \( x \oplus x' = x(x') + x'(x)' = x + x' = 1 \)
(e) \( x \oplus y = xy + x'y = xy + yx' = y \oplus x \)
(f) \( (x \oplus y) \oplus z = (xy + x'y) \oplus z = (xy + x'y)z + (xy' + x'y')z' = xyz + x'y'z + xy'z' + x'y'z' \)
\( = x(x' + y'z') + x'(yz + y') = x(yz + y') + x'(yz + y') = x = (y \oplus z) \)
(g) \( (x \oplus y)' = (xy + x'y')' = (x' + y')(x + y) = x'y + xy' = x \oplus y' = x \oplus y \)

3.19 (a) \( x \oplus y \oplus x = x \oplus [y(xy) + y(xy)] = x \oplus [yx'] = x(x' + x'y) + xy' = x + xy' = x + y \)
(b) \( x \oplus y = xy = xy + x'y = y(x + y) = xy \oplus y = xy + y = x + y \)

3.20 (a) \( xy \oplus xz = xy(x' + z') + (x' + y)xz = xy'z + x'y = xz \)
\( (b) \) For \( y = I \), the left hand side is \( x + z' \) but the right hand side is \( x'z' \) which are not equal.
\( (c) \) For \( y = 0 \), the left hand side is \( xz' \) but the right hand side is \( x' + z' \) which are not equal.
(d) \( (x + y) = (x + z) + (x + y)(x + z)' = x + y + (x'y)(x'z) = x + y + x'y'z' = x + y + z + y'z' \)
\( = x + (y + z) \)

3.21 (a) \( B'C'D' + ABC' + AB'D' + A'BD' + A'BD' = B'C'D' + ABC' + AB'D' + A'BD' = ABC' + AB'D' + A'BD' \)

3.21 (b) \( WY' + WXY + WX'Y + WX = WY' + WXY + WX'Y + WX'Y + WX = WY' + WXY + WX'Y + WX'Y + WX \)
\( = WY' + WXY + WX' \)

3.21 (c) \( (B + C + D) (A + B + C) (A' + C + D) (B' + C' + D') = (A + B + C) (A' + C + D) (B' + C' + D') \)

3.21 (d) \( WXY + WXZ + WY'Z + WZ' = WXY + WXZ + WY'Z + WZ' + XYZ = WY'Z + WZ' + XYZ \)

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Unit 3 Solutions

3.21 (e) \( A'B + B'CD' + \overline{A}D + B'CD + \overline{A}'BD = BC'D' + B'CD + A'BD \)

3.21 (f) \((A + B + C)(B + C' + D)(A' + B' + D') = (A + B + C)(B + C' + D)(A' + B' + D')\)

3.22 \( Z = ABC + DE + ACF + AD'B + AB'E' = A(BC + EF + DE) \)

3.23 \( F = A'B + AC + BC'D' + BDF + BDF = (A + B)(A' + C) + B(C'D' + EF + DF) \)

3.24 \( X'Y'Z' + XYZ = (X + Y'Z') (X' + Y) (X' + Z) (Y + Z') \)

3.25 (a) \( xy + x'y'z + yz = y(x + y')z + yz = xy + yz + y'z \)

3.25 (b) \( (x + y')z = (x + xz + y'z)z \)

3.25 (c) \( xy' + z + (x' + y)z' = x'y + (x' + y) \)  

3.25 (d) \( a'd(b' + c) + a'd'(b + c') + (b' + c)(b + c) \)

3.25 (e) \( w'x' + xy'y + yz + w'z + xz' \)

3.25 (f) \( A'B + AC + BC'D' + BDF + BDF = (A + B)(A' + C) + B(C'D' + EF + DF) \)

3.25 (g) \( xy' + z + (x' + y)z' = x'y + (x' + y) \)
3.25 (f) \[ A'B'CD + A'BC'D + B'E'F + CDE'G + A'D'EF + A'B'EF \]
= \[ A'BD + B'E'F + CDE'G + A'D'EF \] (consensus)
= \[ A'BD + B'E'F + CDE'G \]

3.26 (a) \[ A'CD' + AC' + BCD + A'CD' + A'BC + ABC \]
= \[ A'D' + AC' + BCD + A'BC \] (consensus)
= \[ A'D' + AC' + BCD \]

3.27 \[ WXY'' + (WY' = X) + (Y \oplus WZ) \]
= \[ WXY'' + WY'X + (WY')X' + Y (WZ)' + YWZ \]
= \[ WXY'' + WY'X + (W + Y) X' + Y (W' + Z') +YWZ \]
= \[ XY'' + WX' + XY' + WY' + YZ' + WYZ + WY' \]
= \[ XY'' + WX' + XY' + WY' + YZ' + WYZ + WY' \]
= \[ XY'' + WX' + WY' + YZ' \]
Alternate Solutions: \( F = WY'' + WX' + WZ' + XY' \)
\( F = YZ' + WX' + XY' + WY' \)
\( F = WY'' + WX' + WZ' + WY' \)

3.28 (b) NOT VALID. Counterexample: \( a = 0, b = 1, c = 0 \).
LHS = 0, RHS = 1. \( \therefore \) This equation is not always valid.
In fact, the two sides of the equation are complements:
\[ [(a + b) + (b + c) + (c + b)'] \]
= \[ (a + b) + (b + c) + (c + b)' \]
= \[ (a + b)' + (b + c)' + (b + c) \]

3.28 (d) VALID: LHS = \( xy' + x'z + yz' \)
consensus terms: \( y'z, x'z, xy' \)
= \( xy' + x'z + y'z + x'z + x'y' \)
= \( y'z + x'z + x'y' = RHS \)

3.28 (f) VALID: LHS = \( ab'c + ab'c' + b'c'd + bcd \)
consensus terms: \( ab'd, abd \)
= \( ab'c + ab'c' + b'c'd + bcd + ab'd + abd \)
\[ \frac{ad}{abc' + ab'c + ad + bcd + b'c'd} = RHS \]

3.25 (g) \[ [(a' + d') + (b + d + ac')]' + b'c'd' + a'c'd \]
= \[ ad' (b + c') + b'd' (a' + c') + b'c'd' + a'c'd \]
= \[ abd + ac'd' + a'b'd' + b'c'd' + b'c'd' + a'c'd \]
= \[ abd + b'd' + b'd' + c'd = abd + b'd' + c'd \]

3.26 (b) \[ A'B'C' + ABD + A'C + AC'D + AC'D + AB'C' \]
= \[ B'C' + ABD + A'C + AC'D \]
= \[ B'C' + ABD + A'C \]

3.28 (a) VALID: \( a'b + b'c + c'a \)
= \[ a'b (c + c') + (a + a') b'c + (b + b') ac' \]
= \[ a'b + a'b'c' + ab'c + a'b'c + ab'c' + ab'c' \]
= \[ a'b + ab' + ab'c + ab'c + ab'c' + ab'c' \]
Alternate Solution: \( a'b + b'c + c'a \)
Add all consensus terms: \( ab, bc, ca' \)
\( \therefore \) We get \( ab' + bc' + ca' \)

3.28 (b) NOT VALID. Counterexample: \( x = 0, y = 1, z = 0 \),
then LHS = 0, RHS = 1. \( \therefore \) This equation is not always valid.
In fact, the two sides of the equations are complements:
LHS = \( (x + y) (y + z) (x + z) \)
= \( [(x + y) + (y + z) + (x + z)]' \)
= \( (x' + y' + y'z' + x'z')' = [x'(y' + z') + y'z']' \)
= \( [(x' + y'z') (y' + z' + y'z')]' \)
= \( [(x' + y')(x' + z')(y' + z')]' \)
\# \( (x' + y')(x' + z')(x' + z') \)

3.28 (c) VALID. Starting with the right side, add consensus terms
RHS = \( abd + b'c'd + bc'd + acd + ac'd \)
= \[ abd + b'c'd + bc'd + acd + ac'd \]
= \( abd + b'c'd + bc'd + ad = LHS \)

3.28 (d) VALID: LHS = \( x'y' + x'z + yz' \)
consensus terms: \( y'z, x'z, xy' \)
= \( xy' + x'z + x'y' + y'z + x'z + x'y' \)
= \( y'z + x'z + x'y' = RHS \)

3.28 (e) NOT VALID. Counterexample: \( x = 0, y = 1, z = 0 \),
then LHS = 0, RHS = 1. \( \therefore \) This equation is not always valid.
In fact, the two sides of the equations are complements.
LHS = \( (x + y) (y + z) (x + z) \)
= \( [(x + y) + (y + z) + (x + z)]' \)
= \( (x' + y'z' + x'z')' = [x'(y' + z') + y'z']' \)
= \( [(x' + y'z') (y' + z' + y'z')]' \)
\# \( (x' + y')(x' + z')(y' + z') \)
Unit 3 Solutions

3.29 \( \text{SUM} = (X \oplus Y) \oplus C_3 = (XY' + XY) \oplus C_1 \) 
- \((XY' + XY')C_i \)
- \(XYC_i' + XY'C_i + XYC_i \)
- \(C_0 = (X \oplus Y)C_1 + XY \)
- \(X'Y + X'C_1 + XY \)
- \(X'C_1 + Y'C_1 + XY \)
- \(X_1 + Y'C_1 + XY \)

\[
\begin{array}{cccc|c}
X & Y & C_1 & \text{SUM} & C_0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

3.30 \[
\begin{array}{ccc|c}
A & B & C & F \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\]
\( F = AB + AC + BC \)

3.31 (a) VALID: 
- \( \text{LHS} = (X' + Y) (X \oplus Z) + (X + Y) (X \oplus Z) \)
- \( = (X' + Y) (X'Z' + XZ) + (X + Y) (X'Z' + XZ') \)
- \( = X'Z' + XY'C + XZ' + X'Y'Z' + XZ' + X'Z' \)
- \( = X'Z' + (XY' + X'Y)Z + XZ' \)
- \( = Z' + Z(X \oplus Y) = Z' + (X \oplus Y) = \text{RHS} \)

3.31 (b) \( \text{LHS} = (W' + X + Y) (W + X' + Y) (W + Y + Z) = (W' + X + Y) (W + X' + Y) (WY' + YZ) \)
- \( = W'XY' + WYZ + WX + WY' + X'Y' = W'XY' + WYZ + WX + WY' + X'Y' \)
- \( = W'YZ + WYZ + X'Y' + X'Y' = W'YZ + WX + X'Y' \)

3.31 (c) \( \text{LHS} = ABC + A'C'D' + A'BD' + ACD = AC(B + D) + A'D'(B + C') = (A + D'(B + C')) (A' + C(B + D)) \)
- \( = (A + D')(A + B + C') (A' + C) (B + C' + D) = (A + D')(A' + C) (B + C' + D) \)
- \( \text{RHS} \)

3.32 (a) VALID. \[ A + B = C \] \( \implies [D'(A + B) = D'(C)] \)
\[ A + B = C \] \( \implies [AD' + BD' = CD'] \)

3.32 (b) NOT VALID. Counterexample: \( A = 1, B = C = 0 \) and \( D = 1 \) then \( \text{LHS} = 0(0) + 0(0) = 0 \)
- \( \text{RHS} = 0(1) = 0 = \text{LHS} \)
- \( \text{but } B + C = 0 + 0 = 0; D = 1 \neq B + C \)
- \( \therefore \) The statement is false.

3.32 (c) VALID. \[ A + B = C \] \( \implies [(A + B) + D = (C) + D] \)
\[ A + B = C \] \( \implies [A + B + D = C + D] \)

3.32 (d) NOT VALID. Counterexample: \( C = 1, A = B = 0 \) and \( D = 1 \) then \( \text{LHS} = 0 + 0 + 1 = 1 \)
- \( \text{RHS} = 1 + 1 = 1 = \text{LHS} \)
- \( \text{but } A + B = 0 + 0 = 0 \neq D \)
- \( \therefore \) The statement is false.
A'C' + BC + AB' + A'BD + B'C'D' + ACD'  
Consensus terms:  
(1) B'C' using A'C' + AB'  
(2) A'B' using A'C' + BC  
(3) AC using AB' + BC  
(4) A'BD' using B'C'D' + ACD' 
Using 1, 2, 3:  
A'C' + BC + AB' + A'BD + B'C'D' + ACD' + B'C' + A'B + AC = A'C' + BC + AB'  
(Using the consensus theorem to remove the added terms since the terms that generated them are still present.)

3.33 (b) A'CD' + BC'D + AB'C' + ABC  
Consensus terms:  
(1) A'BC' using A'CD' + BC'D  
(2) AC'D using AB'C' + BC'D  
(3) B'CD' using A'CD' + AB'C'  
(4) A'BD' using A'CD' + A'BC  
(5) A'BD using BC'D + A'BC  
Using 1:  
A'CD' + BC'D + AB'C' + ABC + A'D + B  
which is the minimum solution.

3.34  
abd'f' + b'cegh' + abdf' + acde' + b'ce  
= (abd'f' + abdf') + (b'cegh' + b'ce) + acde'  
= abd'f' + b'ce + acde'  
= abd'f' + b'ce (consensus)  
= (b + ce)(b' + ad')  
= (b + c)(b + e)(b' + a)(b' + d')

3.36  
abc' + d'e + ace + b'c'd'  
= (d' + abc' + ace + b'c'd')(e + abc' + ace + b'c'd')  
= (d' + abc' + ace)(e + abc' + b'c'd')  
= [d' + a(bc' + ce)][e + c'(ab + b'd')]  
= [d' + a(b + c)(c' + e)][e + c'(a + b')(b + d')]  
= (d' + a)(d' + b + c)(d' + c' + e)(e + c')  
(e + a + b')(e + b + d')  
= (d' + a)(d' + b + c)(e + c')  
(e + a + b')(e + b + d')  
= (d' + a)(d' + b + c)(e + c')(e + a + b')  
(consensus)

3.35  
(a + c)(b' + d)(a + c' + d')(b' + c' + d')  
= (a + cd')(b' + c'd)  
= ab' + ac'd + b'cd'

3.37  
(a) (x = y)' = (xy + x'y')' = (x' + y')(x + y)  
= x'y + xy' = x ⊕ y  
(b) a'b'c' + a'bc + ab'c + abc'  
= a'(b'c' + bc) + a(b'c + bc)  
= a'(b = c) + a (b = c)'  
= a' = (b = c)

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Unit 4 Problem Solutions

4.1  See FLD p. 695 for solution.

4.2  

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>

4.3  \[ F_1 = \sum m(0, 4, 5, 6); F_2 = \sum m(0, 3, 4, 6, 7); F_1 + F_2 = \sum m(0, 3, 4, 5, 6, 7) \]

General rule: \( F_1 + F_2 \) is the sum of all minterms that are present in either \( F_1 \) or \( F_2 \).

Proof: Let \( F_1 = \sum a_i m_i \); \( F_2 = \sum b_j m_j \); \( F_1 + F_2 = \sum a_i m_i + \sum b_j m_j = a_0 m_0 + a_1 m_1 + a_2 m_2 + \ldots + b_0 m_0 + b_1 m_1 + b_2 m_2 + \ldots = (a_0 + b_0) m_0 + (a_1 + b_1) m_1 + (a_2 + b_2) m_2 + \ldots = \sum (a_i + b_i) m_i \)

4.4 (a)  \( 2^m = 2^2 = 2^4 = 16 \)

4.4 (b)  

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z_0</th>
<th>z_1</th>
<th>z_2</th>
<th>z_3</th>
<th>z_4</th>
<th>z_5</th>
<th>z_6</th>
<th>z_7</th>
<th>z_8</th>
<th>z_9</th>
<th>z_10</th>
<th>z_11</th>
<th>z_12</th>
<th>z_13</th>
<th>z_14</th>
<th>z_15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

4.5  Alternate Solutions

4.6 (a)  Of the four possible combinations of \( d_1 \) & \( d_5 \), \( d_1 = 1 \) and \( d_5 = 0 \) gives the best solution:

\[ F = A'B'C' + A'B'C + ABC' + ABC = A'B' + AB \]

4.6 (b)  By inspection, \( G = C \) when both don’t cares are set to 0.
4.7 (a) Exactly one variable not complemented: \( F = A'B'C + A'BC' + AB'C' = \sum m(1, 2, 4) \)

4.7 (b) Remaining terms are maxterms:
\[
F = \prod M(0, 3, 5, 6, 7) = (A + B + C)(A + B' + C')(A' + B + C)(A' + B' + C')
\]

4.8

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

4.9 (a) \( F = abc' + b'(a + a')(c + c') = abc' + ab'c + ab'c' + a'b'c + a'b'c' ; F = \sum m(0, 1, 4, 5, 6) \)

4.9 (b) Remaining terms are maxterms: \( F = \prod M(2, 3, 7) \)

4.9 (c) Maxterms of \( F \) are minterms of \( F' \):
\[
F' = \sum M(2, 3, 7)
\]

4.9 (d) Minterms of \( F \) are maxterms of \( F' \):
\[
F' = \prod M(0, 1, 4, 5, 6)
\]

4.10 (a) \( F(a, b, c, d) = (a + b + d)(a' + c)(a' + b' + c')(a + b + c' + d') \)

4.10 (b) \( F = \sum M(1, 5, 6, 7, 11) \)

4.10 (c) \( F' = \sum M(0, 2, 3, 8, 9, 12, 13, 14, 15) \)

4.10 (d) \( F' = \prod M(1, 4, 5, 6, 7, 10, 11) \)

4.11 (a) difference, \( d_i = x_i \oplus y_i \oplus b_i' ; b_{i+1} = b_i x_i' + x_i y_i + b_i y_i \)

4.11 (b) \( d_i = s_i' ; b_{i+1} \) is the same as \( c_{i+1} \) with \( x_i \) replaced by \( x_i' \)

4.13

\[
Z = A'B'C'D' + A'B'C'D + ABC'D + ABCD + A'BCD
\]

\[
= A'B'C' + ABC + A'BC'D' + A'BCD
\]

(Added consensus terms)

\[
\therefore Z = A'B'C' + ABC + BC'D'
\]

4.14

\[
Z = A'B'D + A'BD + A'BD' + A'BD' + A'BD + A'BD
\]

(Added consensus terms)

\[
\therefore Z = AB' + A'BD + A'BC
\]

4.15

(a) Prime digits are 1, 3, 5, and 7 represented as 0010, 0111, 1011 and 1110. The minterms are \(A'B'C'D', A'B'CD, AB'CD\) and \(ABCD\). The don't care minterms are \(A'B'C'D\), \(A'B'CD\), \(A'BC'D\), \(A'BCD'\), \(AB'C'D\) and \(ABCD\).

(b) Nonprime digits are 0, 2, 4, and 6 represented as 0001, 0100, 1000 and 1110. The maxterms are \(A + B + C + D'\), \(A + B' + C + D\), \(A' + B + C + D\) and \(A' + B' + C + D'\). The don't care maxterms are \(A + B + C + D\), \(A + B + C + D'\), \(A + B + C' + D\), \(A' + B + C + D\), \(A' + B + C' + D\) and \(A' + B' + C + D\).

4.16

Truth Table

\[
\begin{array}{c|c|c|c|c|c|c|c|c}
\hline
x_3 & x_2 & x_1 & x_0 & z & y_1 & y_0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & x & x \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
\end{array}
\]

(a) minterms of \(z\): 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

minterms of \(y_1\): 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

don't care minterm: 0

minterms of \(y_0\): 2, 3, 8, 9, 10, 11, 12, 13, 14, 15

don't care minterm: 0

(b) maxterms of \(z\): 0

maxterms of \(y_1\): 1, 2, 3

don't care maxterm: 0

maxterms of \(y_0\): 1, 4, 5, 6, 7

don't care maxterm: 0

4.17

Truth Table

\[
\begin{array}{c|c|c|c|c|c|c|c|c}
\hline
A & B & C & D & W & X & Y & Z \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
\end{array}
\]

(a) minterms of \(w\): 0, 1

minterms of \(x\): 2, 3, 4, 5

minterms of \(y\): 2, 3, 6, 7

minterms of \(z\): 0, 2, 4, 6, 8

don't care minterms: 10, 11, 12, 13, 14, 15

(b) maxterms of \(w\): 2, 3, 4, 5, 6, 7, 8, 9

maxterms of \(x\): 0, 1, 6, 7, 8, 9

maxterms of \(y\): 0, 1, 4, 5, 6, 7, 8, 9

maxterms of \(z\): 1, 3, 5, 7, 9

don't care maxterms: 10, 11, 12, 13, 14, 15
The buzzer will sound if the key is in the ignition switch and the car door is open, or the seat belts are not fastened.

\[ B = K D + S' \]
\[ \therefore \] The two possible interpretations are:
- \[ B = K (D + S') \]
- \[ B = K D + S' \]

You will gain weight if you eat too much, or you do not exercise enough and your metabolism rate is too low.

\[ W = (F + E') M \]
\[ \therefore \] The two possible interpretations are:
- \[ W = F + E'M \]
- \[ W = (F + E') M \]

The speaker will be damaged if the volume is set too high and loud music is played or the stereo is too powerful.

\[ D = VM + S \]
\[ \therefore \] The two possible interpretations are:
- \[ D = V M + S \]
- \[ D = V M' \]

The roads will be very slippery if it snows or it rains and there is oil on the road.

\[ V = S R \]
\[ \therefore \] The two possible interpretations are:
- \[ V = S R \]
- \[ V = S R' \]

\[ Z = AB + AC + BC \]

\[ Z = (A'B'C'D'E')' \]
\[ Y = A'B'CD'E' \]

\[ 13_{10} = D_{16} = 0001101; \therefore X = A'B'C'D'E'G \]
\[ 10_{10} = 0001010; \therefore Y = A'B'C'D'E'F'G' \]

\[ 0_{10} = 0000000; 64_{10} = 1000000; 31_{10} = 0011111; 127_{10} = 1111111; 32_{10} = 0100000; \therefore Z = (A'B')' = A + B \]

\[ F_1 F_2 = \Pi M(0, 4, 5, 6, 7). \text{ General rule: } F_1 F_2 \text{ is the product of all maxterms that are present in either } F_1 \text{ or } F_2. \]

Proof:
Let \( F_1 = \Pi (a_i + M_i); F_2 = \Pi (b_j + M_j); F_1 F_2 = \Pi (a_i + M_i) \Pi (b_j + M_j) \)
\[ = (a_0 + M_0) (b_0 + M_0) (a_1 + M_1) (b_1 + M_1) (a_2 + M_2) (b_2 + M_2) \]
\[ \cdots = (a_0 b_0 + M_0) (a_1 b_1 + M_1) (a_2 b_2 + M_2) \]
\[ \cdots = \Pi (a b + M) \]

Maxterm \( M \) is present in \( F_1 F_2 \) iff \( a b = 0 \), i.e., if either \( a = 0 \) or \( b = 0 \). Maxterm \( M \) is present in \( F_1 \) iff \( a = 0 \). Maxterm \( M \) is present in \( F_2 \) iff \( b = 0 \). Therefore, maxterm \( M \) is present in \( F_1 F_2 \) if it is present in \( F_1 \) or \( F_2 \).
4.24 
\[ F_1 + F_2 = \prod M(0, 4) \]. General rule: \( F_1 + F_2 \) is the product of all maxterms that are present in both \( F_1 \) and \( F_2 \).

Proof:
Let \( F_1 = \sum_{i=0}^{2^4-1} (a_i m) \); \( F_2 = \sum_{i=0}^{2^4-1} (b_i m) \); \( F_1 + F_2 = \sum_{i=0}^{2^4-1} (a_i m) + \sum_{i=0}^{2^4-1} (b_i m) \)

\[ = a_i m_n + b_i m_n + a_i m_1 + b_i m_1 + a_i m_2 + b_i m_2 \ldots \ldots = (a_n + b_n) m_n + (a_1 + b_1) m_1 + (a_2 + b_2) m_2 \ldots \ldots \]

Minterm \( m_i \) is present in \( F_1 + F_2 \) iff \( a_i + b_i = 1 \), i.e., if either \( a_i = 1 \) or \( b_i = 1 \) so maxterm \( M_i \) is present in \( F_1 + F_2 \) if \( a_i = 0 \) and \( b_i = 0 \). Therefore, maxterm \( M_i \) is present in \( F_1 + F_2 \) if it is present in both \( F_1 \) and \( F_2 \).

4.25

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) \( F(A, B, C, D) = \sum m(5, 6, 7, 10, 11, 13, 14, 15) \)

(b) \( G(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 6, 8, 9, 12) \)

(c) \( H(A, B, C, D) = \sum m(7, 11, 13, 14, 15) \)

(d) \( J(A, B, C, D) = \sum m(4, 8, 12, 13, 14) \)

4.26

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) \( F(A, B, C, D) = \sum m(5, 7, 10, 11, 13, 14, 15) \)

(b) \( G(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 6, 8, 9, 12) \)

(c) \( H(A, B, C, D) = \sum m(7, 11, 13, 14, 15) \)

(d) \( J(A, B, C, D) = \sum m(4, 8, 12, 13, 14) \)

4.27

You can also work this problem using a truth table, as in problem 4.28.

\( f(a, b, c) = a(b + c') = ab + ac' = ab(c + c') + a(b + b')c' = abc + abcc' + abcc' + abcc' = \prod m_1 m_6 m_6 m_4 \)

\( f = \sum m(4, 6, 7) \) \( f = \prod M(0, 1, 2, 3, 5) \)

\( f' = \sum m(0, 1, 2, 3, 5) \) \( f' = \prod M(4, 6, 7) \)

4.28

\( f(a, b, c) = a(b + c') = ab + ac' = \sum m(0, 1, 2, 3, 5) \)

(a) \( f = \sum m(1, 2, 4, 5, 6, 11, 12, 14, 15) \)

(b) \( f = \prod M(0, 3, 7, 8, 9, 10, 13) \)

(c) \( f' = \sum m(0, 3, 7, 8, 9, 10, 13) \)

(d) \( f'' = \prod M(1, 2, 4, 5, 6, 11, 12, 14, 15) \)

You can also work this problem algebraically, as in problem 4.27.
4.29 (a) \[ f(A, B, C, D) = AB + A'CD + A'B'CD' + ABC'D' + ABC'D \]
\[ + A'B'CD + A'B'CD + A'BCD + A'BC'D + ABC'D \]
\[ = (A + A'D')(B + A'CD) = (A + C)(A' + D)(A' + B) \]
\[ (B + C)(B + D) \]
\[ f(A, B, C, D) = (A + B' + C + D')(A + B + C + D) \]
\[ (A + B + C + D')(A + B + C + D)(A + B + C + D) \]
\[ (A' + B' + C + D')(A' + B + C' + D)(A' + B + C + D) \]
\[ (A' + B' + C + D')(A' + B' + C' + D)(A' + B' + C + D') \]
\[ (A' + B' + C + D)(A + B + C + D) \]
\[ Note: Consensus could have been applied twice to write \( f = (A + C)(A + D)(A' + B) \) and save some work. \]

4.30 (a) \[ F(A, B, C, D) = \sum m(3, 4, 5, 8, 9, 10, 11, 12, 14) \]
\[ F = A'B'CD + A'BC'D' + A'B'C'D + A'BC'D + A'B'CD' + A'B'CD + ABC'D + A'B'CD' + ABC'D + ABCD' \]

4.31 (a) \[ F(A, B, C, D) = \sum m(0, 3, 4, 7, 8, 9, 11, 12, 13, 14) = A'B'CD' + A'B'CD' + A'BC'D' + A'B'C'D' + A'B'CD' + A'B'CD' + ABCD' \]
\[ + A'B'CD + A'B'CD' + ABCD + ABCD' \]
\[ m_0 \quad m_1 \quad m_2 \quad m_3 \quad m_4 \quad m_5 \quad m_6 \quad m_7 \quad m_8 \quad m_9 \]

4.31 (b) \[ F(A, B, C, D) = \prod M(1, 2, 5, 6, 10, 15) = (A + B + C + D')(A + B + C + D)(A + B' + C + D')(A + B' + C + D)(A + B' + C + D)' \]
\[ (A' + B + C + D)(A' + B + C + D) \]

4.32 (a) If don't cares are changed to (1, 1), respectively, \[ F_1 = A'B'C' + ABC + A'B'C + AB'C \]
\[ = A'B' + AC \]

4.32 (c) If don't cares are changed to (1, 1), respectively \[ F_3 = (A + B + C)(A + B + C) = A + B \]

4.33

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>F</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X²</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

1 These truth table entries were made don't cares because \( ABC = 110 \) and \( ABC = 011 \) can never occur.

2 These truth table entries were made don't cares because when one input of the OR gate is 1, the output will be 1 regardless of the value of its other input.

4.34 (a) \[ G_1(A, B, C) = \sum m(0, 7) = \prod M(1, 2, 3, 4, 5, 6) \]
4.34 (b) \[ G_2(A, B, C) = \sum m(0, 1, 6, 7) = \prod M(2, 3, 4, 5) \]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
4.35 (a)  

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>1's</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

4.35 (b)  

\[ X = A'B'C'D + A'B'CD' + A'B'CD + A'B'C'D' + AB'CD' + AB'CD + ABC'D' + ABC'D + ABCD + ABCD' + 1 \]

\[ Y = (A + B + C + D)(A + B' + C + D') \]
\[ Z = (A + B + C + D)(A + B' + C + D') \]
\[ (A' + B + C + D')(A' + B' + C' + D') \]

4.36 (a)  

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

4.36 (b)  

\[ X = A'B'C'D + A'B'CD' + A'B'CD + A'B'C'D' + AB'CD' + AB'CD + ABC'D' + ABC'D + ABCD + ABCD' + 1 \]

\[ Y = (A + B + C + D')(A + B + C' + D) \]
\[ Z = (A + B + C + D)(A + B' + C + D') \]
\[ (A' + B + C' + D')(A' + B' + C + D') \]

4.37  

\[ AB CD \times SUV \times WX Y Z \]
\[ 0 \times 5 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]

4.38  

\[ AB CD \times SUV \times WX Y Z \]
\[ 0 \times 5 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]

**Note:** Rows 1010 through 1111 have don't care outputs.

\[ S = 0, T = A, U = B, V = C, W = 0, X = D, Y = 0, Z = D \]

**Unit 4 Solutions**

4.37  

\[ AB CD \times SUV \times WX Y Z \]
\[ 0 \times 5 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]

4.38  

\[ AB CD \times SUV \times WX Y Z \]
\[ 0 \times 5 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]
\[ 0 \times 1 = 0 \]

**Note:** Rows 1010 through 1111 have don't care outputs.

\[ S = 0, T = 0, U = BD + BC + A, V = B'CD + BCD' + A, W = B'CD' + BCD, X = B'CD + BD', Y = B'CD + BCD' + A, Z = 1 \]
Unit 4 Solutions

4.39 Notice that the sign bit $X_3$ of the 4-bit number is extended to the leftmost full adder as well.

4.40

4.41 (a), (b), (c) $f = x(y+y') + y(x+x') = xy + xy' + x'y$

(b) $f = x + y$ already in product-of-maxterms form

(c) $f' = (a' + x')(b' + y') = (b + x')(a + y')$

4.42 (a) $m_1 + m_2 = m_1(m_2' + m_2) + (m_1' + m_1)m_2$

(b) Using part (a), any function can be written as the exclusive-or sum of its minterms. However, if a product contains a complemented literal, it can be written as the exclusive-or sum of two products without a complemented literal by using

$$x'p = (x \oplus 1)p = xp \oplus p.$$
Unit 5 Problem Solutions

5.3 (a) \[ f_1 = a'c' + a b'c + b c' \]

5.3 (b) \[ f_2 = d'e' + d'f' + c'f' \]

5.3 (c) \[ f_3 = r' + t' \]

5.3 (d) \[ f_4 = x'z + y + x z' \]

5.4 (a) \[ F = (A + B' + D')(B + C + D') \]

5.4 (b) \[ F = BD' + B'CD + ABC + ABC'D + B'D' \]

5.4 (c) \[ F = D' + B'C + A B \]

5.5 (a) See FLD p. 697 for solution.

5.5 (b) \[ C_1 C_2 \]

\[ Z = C_1 X_1 X_2 + C_1 C_2 X_1 X_2 + C_1 X_1 X_2 + C_1 C_2 X_2 \\
Alt: Z = C_1 X_1 X_2 + C_1 C_1 X_1 X_2 + C_1 C_2 X_1 X_2 + C_1 C_2 X_2 \]

5.6 (a) \[ f = a'b'c' + a'd + b'cd + abd' + bcd' \\
Alt: f = a'b'c' + a'd + b'cd + abd' + a'bc \]

(*) Indicates a minterm that makes the corresponding prime implicant essential.

a'd \[ \rightarrow m_4 \]; a'b'c' \[ \rightarrow m_5 \]; b'cd \[ \rightarrow m_{11} \]; abd' \[ \rightarrow m_{12} \]

5.6 (b) \[ F = a'c + b'd' + bd + a'd' \\
Alt: F = a'c + b'd' + bd + a'b \]

(*) Indicates a minterm that makes the corresponding prime implicant essential.

bd \[ \rightarrow m_{13} \] or \[ m_{13} \]; a'c \[ \rightarrow m_3 \]; b'd' \[ \rightarrow m_9 \] or \[ m_{10} \]
Unit 5 Solutions

5.6 (c)

\[ F = a'd'd + b' + c'd' \]

(\(^*)\) Indicates a minterm that makes the corresponding prime implicant essential.

c'd'→m\(_{12}\); a'd'→m\(_{10}\); b'→m\(_{10}\) or m\(_{11}\)

5.7 (b)

\[ f = a'b' + a'c'd' + abc \]

5.7 (c)

\[ f = a'c'd' + a'cd + b'c'd' + abcd' + a'b'c \]

Alt: \[ f = a'c'd' + a'cd + b'c'd' + abcd' + a'b'c \]

5.7 (d)

\[ F = D + A C \]

5.8 (a)

\[ f = (c'+ d')(b' + c')(a + b + c)(a' + c + d) \]

5.8 (b)

\[ f = (a' + c)(b' + d')(b + d)(c' + d) \]

Alt: \[ f = (a' + c)(b' + d')(b + d)(b' + c') \]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
5.9 (a)


5.9 (b)

\[ F = (A' + B' + E)(A' + C' + D + E)(C + D' + E')(A + B + D' + E')(A + B + C)(B + D + E') \]

5.10 (a)

Essential prime implicants: \( c'd'e'(m_{16}, m_{24}) \), \( a'c'e'(m_{14}) \), \( ace(m_{31}) \), \( a'b'de(m_{1}) \)

5.10 (b)

Prime implicants: \( a'bde, a'd'ei, cd'e, ace', ace, a'b'c, b'ce, c'd'e', a'cd' \)
5.11

\[ f = (a' + b + c') (a' + d' + e) (a + b' + c') (a + c + e') \]

Alt: \[ f = (d' + b + c') (a' + d' + e) (a + b' + c') (a + c + e') \]

5.12 (a)

\[ F = A \overline{B} D' + A \overline{B} + A \overline{C} + \overline{C} D \]

\[ F = \prod M(0, 1, 9, 12, 13, 14) = (A + B + C + D) \]
\[ (A + B + C + D') (A' + B' + C + D) \]
\[ (A' + B' + C + D') (A' + B' + C' + D) \]
\[ (A' + B + C + D') \]

5.12 (b)

\[ F' = A' B' C' + A B D' + A C D \]

5.12 (c)

\[ F = (A' + B' + D) (A + B + C) (A' + C + D') \]

5.13

Minterms \( m_1, m_4, m_9, m_{12}, m_{13}, m_{16}, m_{19}, \) and \( m_{14} \) can be made don't cares, individually, without changing the given expression. However, if \( m_{13} \) or \( m_{14} \) is made a don't care, the term \( B C'D \) or the term \( A C D' \) (respectively) is not needed in the expression.

5.14 (a)

\[ f_1 = B'C + A'BC + AC \]

5.14 (b)

\[ f_2 = e'f + d e \]

5.14 (c)

\[ f_3 = s' + f \]

5.14 (d)

\[ f_4 = a'c' + be \]
5.14 (e) \[ f_5 = n'q + np' \]

5.14 (f) \[ f_6 = z' + x'y + xy' \]

5.15 (c) \[ f_3 = (s'+ t') \]

5.15 (d) \[ f_4 = (b + c')(a'+ c) \]

5.15 (e) \[ f_5 = (n + q')(n' + p') \]

5.15 (f) \[ f_6 = (x + y + z')(x' + y' + z') \]

5.16 (a) \[ f_1 = A'C + AC' \]

5.16 (b) \[ f_2 = e'f + de' + df \]

5.16 (c) \[ f_3 = t' + r \]

5.16 (d) \[ f_1 = bc + ac' \]

5.16 (e) \[ f_2 = n'q + np' \]

5.16 (f) \[ f_4 = y' + x'z' + xz \]

5.17 (a) & (b) \[ F = A'B' + CD' + ABC \]
5.20 (e) \[ f = a'b' + a b + a c \]

5.20 (f) \[ G = D E F + D'E' \]
\[ G = D E F + D'F \]
\[ G = D E F + E'F \]
Unit 5 Solutions

5.21

F = a'b'c' + a'c'd + bcd + abc + a b'
= (a'b'c' + ab') + a'c'd + bcd + (abc + a b')
= (c' + a)b' + (a'c'd + bcd) + (abc + b')
= (b'c' + ac + ab') + a'bd
= b'c' + ac + a'bd

5.22 (a)

f = A B' + B D' + A C' or
= A B' + B C' + B D' or
= A B' + B C' + A D'
PIs: A B', B C', A D', B D', A C', A B'

5.22 (b)

f = B'C D + A C' + A D'
PIs: B'C D, A C', A D', A B', B C'D, B C D', A'C D, A'B D, A'B C

5.22 (c)

f = C'D + C D' + A'B
PIs: C'D, C D', A'B, A B'C', A B'D'

5.22 (d)

f = A'B + B C D
PIs: A'B, B C D, A B'C', A B'D'

5.22 (e)

f = B D'
PIs: A'B, B C', B D', A C', A D', A B'

5.22 (f)

f = B C D + A'C' or
f = B C D + A'D' or
f = B C D + A'C'
PIs: B C D, A'C', A'D', A'B, B'D', B'C'

5.23 (a)

f = (B'+ C'+ D') (A + B )
PIs: (B'+ C'+ D'), (A + B ), (A' + B), (A + B' + D'), (A + B' + C')

5.23 (b)

f = (B + C + D ) (C' + D') (A'+ C + D )
PIs: (B + C + D ), (C' + D'), (A'+ C + D ), (A' + B), (A + B' + D'), (A + B' + C')

5.23 (c)

f = (B + C + D ) (C' + D') (A'+ C + D )
PIs: (B + C + D ), (C' + D'), (A'+ C + D ), (A' + B), (A + B' + D'), (A + B' + C')

5.23 (d)

f = (B ) (A'+ C ) (A'+ D )

Unit 5 Solutions

5.23 (e)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pls: (C + D'), (A'+ D), (B + D), (A' + C), (B + C), (C' + D), (A + B' + D'), (A + B' + C)

f = (A' + C ) (B + C ) (C' + D ) or
   = (C + D') (A' + D ) (B + D)

Alt: F = A B C + B'C D + A'C + A'B'D' + A'B D

5.24 (a)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F = A B C' + B'C D + A'C + A'B'D' + A'B D

Alt: F = A B C + B'C D + A'C + A'B'D' + B C'D

5.24 (b)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F = A'C'D' + B'C'D' + A'B'C

Alt: F = A'C'D' + B'C'D' + A'B'D'

5.24 (c)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F = A'C'D + A'B + B C'D

5.24 (d)

<table>
<thead>
<tr>
<th></th>
<th>w</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

f = x'y' + w'z + y'z + wz'

Alt: \[ f = x'y' + wy' + wz + wz' \]

5.24 (e)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

f = (B ) (A') or
   = (B ) (C') or
   = (B ) (D')

Pls: (B)(A'+ D)(A'+ C)

5.25 (a)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

f = a'd + a'bc' + c'd + bd

5.25 (b)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

f = b'c'd + cd' + bd' + bc + ab

5.25 (c)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

f = b'd' + cd' + a'c'd

5.25 (d)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

f = a'be' + ab'd' + ac

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
5.27 (a)  
\[ f = (A' + B' + D)(A' + B')(A + B) \]

5.27 (b)  
\[ f = x'y' + w'z + y'z + wz' \]

5.28  
\[ F = b'd' + a'd + c'd \]

Notice that abcd = 0101 and 1111 never occur, so minterms 5 and 15 are don't cares.

5.29 (a)  
\[ F = A'B'D' + A'B + A'C + C \]

5.29 (b)  
\[ F = ABD' + A'B'C' + AC'D \]

\[ F = \prod M(0, 1, 9, 12, 13, 14) = (A + B + C + D)(A + B + C + D') \]
\[ (A' + B + C + D')(A' + B' + C + D) \]
\[ (A' + B' + C + D')(A' + B' + C' + D) \]
### Unit 5 Solutions

#### 5.29 (c)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ F = (A' + B' + D)(A + B + C)(A' + C + D') \]

#### 5.30

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

\[ F = D + AB C \]

#### 5.31

Prime implicants for \( f' \): \( abc'e, ac'd', ab'e, a'ce, b'c'd'e, c'd'e, a'd'e \)

Prime implicants for \( f \): \( a'd'e', ace, a'ce', bde', abc, bce', b'c'd'e, a'b'c'd'e, ab'd'e \)

#### 5.32

For \( F \): \( b'c'd'e', a'ce, ab'e', ac'd', abc'e, c'd'e, a'd'e \)

For \( G \): \( ab'c'e, a'bc'd, ab'd'e, cde, b'de, a'b'c'd, a'c'e' \)

#### 5.33

5-variable mirror image map

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Essential Pls: \( ab'c'e, ab'ce, ab'ed \)

\[ f = ab'c'd' + ab'ce + ab'ed + ab'c'd' + ac'd'e + ac'd'e' \]

Other Pls: \( ab'd', b'c'd', b'c'd'e, a'b'd'e, b'c'd'e \)

#### 5.34 (a)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Pls: \( bd'e', ab'b, ab'd', ac, ab'd \)

\[ f = b'd' + c \text{ or } a'b + c \text{ or } a'd' + c \]

#### 5.34 (b) & (c)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Pls: \( bd'e', ab'b, ab'd', c, ab'd \)

\[ f = b'd' + c \text{ or } a'b + c \text{ or } a'd' + c \]

#### 5.34 (d) & (e)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Pls: \( c + d', (a' + c), (a + b + d'), (a + b + c + d'), (a' + b + d'), (a' + b + d) \)

\[ f = (c + d')(a' + c) \text{ or } (b + c)(c + d') \text{ or } (b + c)(a' + c) \]
5.35 (a), (b) & (c) 5-variable mirror image map

Pls: A, C'D', B'E, C E, B D', D'E, B C'E', B'C D
F = A + B'E + BD' or
= A + C'D' + CE

5.35 (d), (e) 5-variable mirror image map

Pls: (A'+ B'), (A'+ C'), (A'+D + E'), (B'+D'), (B'+ C + E'),
(C'+ E), (D'+E), (B + C'+ D), (A + C + D'), (A + B + E)
F = (B'+D')(A + B + E) or
= (C'+ E)(A + C + D')

5.36 (a), (b) & (c) 5-variable mirror image map

Pls: A, B, A D, A C E, B C, B D'E, C D, D E, C E',
A'C, B'D, C'D'E, A'D'E, B'C'E, A'B'
F = A'C + B'D + AB or
= B'D + AB + BC or
= A'C + AB + AD

5.36 (d) 5-variable mirror image map

Pls: (A'+ B'+ C'), (A'+B'+D'), (A'+ B+E), (A'+ C+E'),
(A'+C+ D), (B'+ D'+ E), (B'+C+ D'), (D + E),
(A + B + E), (A + C), (B + D ), (C + E)
F = (A + C)(B + D )
5.37 (a), (b) & (c)

\[ f = a'b'c + ab'c' + ab'd' + bc'de + a'b'd' + bc'd'e \]


F = A'B'E + A'B'D' + A B'C' or
    = ACD' + A'C E + A B'C' or
    = A'D'E + B'D E + C'DE' or
    = A'B D' + B'C'D' + B'D E or
    = A'C E + B'C'E + C'D'E'

(*) Indicates a minterm that makes the corresponding prime implicant essential.

a'b'd'→m₁; cd'e'→m₁₈; bc'd'e→m₂₅; b'cd'→m₂₁

5.38 (a)

\[ f = a'b'c + ab'c' + ab'd' + bc'de + a'b'd' + bc'd'e \]

Alt: \[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

5.38 (b)

\[ f = a'b'c + ab'c' + ab'd' + bc'de + a'b'd' + bc'd'e \]

5.39 (a)

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

Alt: \[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

5.39 (b)

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

Alt: \[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]

\[ f = a'b'c + ab'c' + ab'd' + c'd'e' + ab'd + bde + a'e' \]
5.40

\[ F = A'B'CD' + BC'D' + BCD' + BCD'E' + ABC'D \]

5.41

\[ F = AB'CD'E' + BC'D' + ABDE' + AB'C'E' + AC'D'E + ACD'E \]

5.42 (a)

\[ F = VYXZ + X'YZ + VZ + WX'YZ + VWX \]

5.42 (b)

\[ F = (X + Y + Z) (V + Y' + Z') (V + X' + Z') \]
\[ (V' + X' + Y') (V' + W + Z) \]

5.43 (a)

\[ F = (c + d + e)(a' + b)(a + b')(a + c' + d') + c' \]

Alt: \[ F = (c + d + e)(a' + b)(a + b')(a + c' + d') + c' \]

5.43 (b)

\[ F = (c + d')(a + d' + e')(a + b' + c)(b' + c + e') \]
\[ (a' + b' + c' + d)(b + c + e) \]

Alt: \[ F = (c + d')(a + d' + e')(a + b' + c)(b' + c + e') \]
\[ (a' + b' + c' + d)(b + c + e) \]
Unit 5 Solutions

5.44 (a)

\[ F = (v' + w' + x' + y' + z')(w + y' + z')(v + y')(w + x + y) \]
\[ (v' + x + y + z)(w + x + y') \]

Alt:\n\[ F = (v' + w' + x' + y + z')(w + y' + z')(v + y')(w + x + y) \]
\[ (v' + w' + x + z)(w' + x + y') \]

5.45 (a)

\[ F = ACD' + BC'D + B'C + A'C' \]

m₄, m₁₃, or m₁₄ change the minimum sum of products, removing A'C', BC'D, or ACD', respectively.

5.45 (b)

\[ F = CD + BD + AB' \]

Changing \( m₁ \) to a don't care removes \( C'D \) from the solution.

5.46 (a)

\[ F = V'XY + V'WZ' + XYZ + V'WXY' + VWY'Z' + W'XZ \]
\[ F = V'XY' + V'WZ' + XYZ + V'WXY' + VWY'Z' + WY'Z \]
\[ F = V'XY + V'WZ' + XYZ + VWXY' + VWY'Z' + WY'Z \]
\[ F = V'XY + V'WZ' + XYZ + VWXY' + VWY'Z' + WYZ \]

5.46 (b) \( V'WZ' \to m₈; XYZ \to m₁₃; V'XY' \to m₄ \)
6.2 (a) Prime implicants: $a'c'd, b'c'd, a'bd, ab'd$, $abd'$, $bcd, acd, abc$

6.2 (b) Prime implicants: $a'b'c', b'c'd', ab'd'$, $acd'$, $a'd$, $bc$

6.3 (a) $f = abd' + a'c'd + ab'd + bcd$

6.3 (b) $f = a'd + bc + a'b'c' + ab'd'$
### Unit 6 Solutions

#### 6.4

<table>
<thead>
<tr>
<th></th>
<th>1 0001</th>
<th>1, 3</th>
<th>0-1</th>
<th>1, 3, 5, 7</th>
<th>0--1</th>
<th>a'd</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0010</td>
<td>1, 5</td>
<td>0-1</td>
<td>1, 5, 2, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>1, 9</td>
<td>0-01</td>
<td>1, 5, 9, 13</td>
<td>--01</td>
<td>c'd</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>2, 3</td>
<td>001-</td>
<td>2, 3, 6, 7</td>
<td>0--1</td>
<td>a'e</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>2, 6</td>
<td>0-10</td>
<td>2, 3, 6, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>2, 10</td>
<td>010-</td>
<td>2, 6, 3, 7</td>
<td>0--1</td>
<td>b'c'</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>4, 5</td>
<td>010-</td>
<td>4, 5, 6, 7</td>
<td>01--</td>
<td>a'b</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>4, 6</td>
<td>01-0</td>
<td>4, 5, 12, 13</td>
<td>-10-</td>
<td>b'c'</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>4, 12</td>
<td>0-10</td>
<td>4, 6, 5, 7</td>
<td>01--</td>
<td>b'd</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>3, 7</td>
<td>0-11</td>
<td>4, 12, 5, 13</td>
<td>-10-</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>5, 7</td>
<td>01-1</td>
<td>5, 7, 13, 15</td>
<td>-1-1</td>
<td>b'd</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>5, 13</td>
<td>-101</td>
<td>5, 13, 7, 15</td>
<td>-1-1</td>
<td></td>
</tr>
</tbody>
</table>

Prime implicants: \( b'c'd', a'd, c'd, a'c, a'b, bc', bd \)

#### 6.5

<table>
<thead>
<tr>
<th></th>
<th>1 0001</th>
<th>1, 5</th>
<th>0-1</th>
<th>1, 5, 9, 13</th>
<th>--01</th>
<th>CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0100</td>
<td>1, 9</td>
<td>0-01</td>
<td>1, 9, 5, 13</td>
<td>0--1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>4, 5</td>
<td>010-</td>
<td>4, 5, 12, 13</td>
<td>-10-</td>
<td>BC</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>4, 12</td>
<td>0-10</td>
<td>4, 12, 5, 13</td>
<td>0-10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>8, 9</td>
<td>0100</td>
<td>5, 7, 13, 15</td>
<td>-1-1</td>
<td>BD</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>8, 12</td>
<td>1-00</td>
<td>8, 12, 7, 15</td>
<td>1-1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>5, 7</td>
<td>01-1</td>
<td>8, 9, 12, 13</td>
<td>1-0-</td>
<td>AC</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>5, 13</td>
<td>-101</td>
<td>8, 12, 9, 13</td>
<td>1--0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>9, 11</td>
<td>10-1</td>
<td>9, 11, 13, 15</td>
<td>1--1</td>
<td>AD</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>9, 13</td>
<td>1-01</td>
<td>9, 13, 11, 15</td>
<td>1--1</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>12, 13</td>
<td>110-</td>
<td>12, 13, 14, 15</td>
<td>11--</td>
<td>AB</td>
</tr>
</tbody>
</table>

Prime implicants: \( C'D, BC', BD, AC', AD, AB \)

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
### 6.5 (contd)

<table>
<thead>
<tr>
<th></th>
<th>9</th>
<th>12</th>
<th>13</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 (1, 5, 9, 13)</td>
<td>CD</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>P2 (4, 5, 12, 13)</td>
<td>BC</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>P3 (5, 7, 13, 15)</td>
<td>BD</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>P4 (8, 9, 12, 13)</td>
<td>AC</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P5 (9, 11, 13, 15)</td>
<td>AD</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P6 (12, 13, 14, 15)</td>
<td>AB</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

\[ \begin{align*}
(P1 + P4 + P5) & = (P2 + P4 + P6) \quad (P1 + P2 + P3 + P4 + P5 + P6) \quad (P3 + P5 + P6) \\
& = (P4 + P1P2 + P1P6 + P2P5 + P5P6) \quad (P3 + P5 + P6) \\
& = P3P4 + P4P5 + P4P6 + P1P2P3 + P1P2P5 + P1P2P6 + P1P3P6 + P1P5P6 + P1P6 + P2P3P5 + P2P5P6 + P2P5P6 + P5P6 = 1 \\
\end{align*} \]

\[ F = (AC' + BD) \text{ or } (AD + BC') \text{ or } (AD + AC') \text{ or } (AB + AD) \text{ or } (AB + AC') \text{ or } (AB + C'D) \]

### 6.6 (a)

\[ F = M_{S_0} + E \times M_{S_1} = A'B + A'CD' + ABD + E \times (AC' + ACD) \]

or \[ E \times (AC' + BCD) \]

\[ M_{S_0} = A'CD' + A'B + ABD \\
M_{S_1} = A'C + ACD \]

### 6.6 (b)

\[ F = 1; E = G = 0 \]

\[ M_{S_0} = A'B + ABD \]

\[ M_{S_1} = BC + AC \]

\[ Z = A'BD + E \times (BC' + AC' + F(AB)) + G(A'D) \]

\[ Z = A'BD + E \times (BC' + AC' + F(AB)) + G(A'D) \]

\[ MS_2 = A'B \]

\[ MS_3 = A'D \text{ or } CD \text{ or } BD \]
Unit 6 Solutions

6.7 (a) 

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0000 ✓</th>
<th>0, 4 0-0-0 a'c'd'</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>0100 ✓</td>
<td>4, 5 010- a'be'</td>
</tr>
<tr>
<td>3</td>
<td>0011 ✓</td>
<td>3, 7 0-11 a'cd</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0101 ✓</td>
<td>3, 11 -011 b'cd</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1001 ✓</td>
<td>5, 7 01-1 a'bd</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0111 ✓</td>
<td>5, 13 101 b'cd'</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1011 ✓</td>
<td>9, 11 10-1 ab'd</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1101 ✓</td>
<td>9, 13 1-01 ac'd</td>
<td></td>
</tr>
</tbody>
</table>

Prime implicants: a'c'd', a'be', a'cd, b'cd, a'bd, bc'd, ab'd, ac'd

---

6.7 (b) 

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>0010 ✓</th>
<th>2, 6 0-10 a'cd'</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0100 ✓</td>
<td>2, 10 -010 b'cd'</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0101 ✓</td>
<td>4, 5 010-</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0110 ✓</td>
<td>4, 6 01-0 a'bd'</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1001 ✓</td>
<td>4, 12 -100</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1010 ✓</td>
<td>5, 13 -101</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1100 ✓</td>
<td>9, 11 10-1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1011 ✓</td>
<td>9, 13 1-01</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1101 ✓</td>
<td>10, 11 101- ab'c</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1111 ✓</td>
<td>12, 13 110-</td>
<td></td>
</tr>
</tbody>
</table>

Prime implicants: a'c'd', a'bd, a'cd, a'be', b'cd, ad, ac'd

---

6.8 (a) 

|   | 0 | 3 | 4 | 5 | 7 | 9 | 11 | 13 |
|---|---|---|---|---|---|---|-----|
| 0, 4 | a'c'd' |
| 4, 5 | a'be' |
| 3, 7 | a'cd |
| 3, 11 | b'cd |
| 5, 7 | a'bd |
| 5, 13 | b'cd |
| 9, 11 | ab'd |
| 9, 13 | ac'd |

6.8 (b) 

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 6</td>
<td>a'cd'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2, 10</td>
<td>b'cd'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4, 6</td>
<td>a'bd'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10, 11</td>
<td>ab'c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4, 5, 12, 13</td>
<td>be'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9, 11, 13, 15</td>
<td>ad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Prime implicants: bc', be', a'cd' + a'cd + ab'd + be'd

---

6.8 (a) 

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 6</td>
<td>a'cd'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2, 10</td>
<td>b'cd'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4, 6</td>
<td>a'bd'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10, 11</td>
<td>ab'c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4, 5, 12, 13</td>
<td>be'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9, 11, 13, 15</td>
<td>ad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Prime implicants: bc', be', a'cd' + a'cd' + b'cd'
6.9 (a) Prime implicants: $bc'd', b'd, b'c, cd, ad, ac, ab$

6.9 (b) Prime implicants: $a'bd, a'be, b'c', c'd, ab', ac'$

Unit 6 Solutions
### 6.9 (c)

\[
f = a'b + be + ab'c' + bd + cd
\]

\[
f = a'b + be + ab'c' + ad + cd
\]

\[
f = a'b + bc + ab'c' + ad + a'c
\]

### 6.10

Prime implicants: \(abc', bc'd, a'bd, b'cd, a'c, a'b'd'

\[
f = abc' + b'cd + a'c + a'b'd' + a'bd
\]

\[
f = abc' + b'cd + a'c + a'b'd' + bc'd
\]

### 6.11

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00000</td>
<td>0, 2</td>
<td>000-0</td>
<td>0, 2, 4, 6</td>
<td>00--0</td>
<td>0, 2, 4, 6, 8, 10, 12, 14</td>
<td>0---0</td>
</tr>
<tr>
<td>2</td>
<td>00010</td>
<td>0, 4</td>
<td>00-0</td>
<td>0, 2, 8, 10</td>
<td>0-0-0</td>
<td>0, 2, 8, 10, 4, 6, 12, 14</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>00100</td>
<td>0, 8</td>
<td>0-00</td>
<td>0, 2, 16, 18</td>
<td>-00-0</td>
<td>0, 4, 8, 12, 6, 10, 14</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>01000</td>
<td>0, 16</td>
<td>-0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10000</td>
<td>2, 6</td>
<td>00-10</td>
<td>0, 4, 8, 12</td>
<td>0-00</td>
<td>0, 4, 8, 12, 6, 10, 14</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>10000</td>
<td>2, 10</td>
<td>0-10</td>
<td>0, 8, 2, 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>01001</td>
<td>2, 18</td>
<td>-0010</td>
<td>0, 8, 4, 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>01010</td>
<td>4, 6</td>
<td>001-0</td>
<td>0, 16, 2, 18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>01100</td>
<td>4, 12</td>
<td>0-100</td>
<td>2, 6, 10, 14</td>
<td>0--10</td>
<td>2, 10, 6, 14</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>00111</td>
<td>8, 10</td>
<td>010-0</td>
<td>4, 6, 12, 14</td>
<td>0-1-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>01011</td>
<td>8, 12</td>
<td>01-00</td>
<td>4, 12, 6, 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>01101</td>
<td>16, 18</td>
<td>100-0</td>
<td>8, 9, 10, 11</td>
<td>010--</td>
<td>A'BC'</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>01110</td>
<td>6, 7</td>
<td>0011- A'B'DC</td>
<td>8, 9, 12, 13</td>
<td>01-0-</td>
<td>A'B'D</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>10011</td>
<td>6, 14</td>
<td>0-110</td>
<td>8, 10, 9, 14</td>
<td>011-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>10101</td>
<td>9, 11</td>
<td>010-1</td>
<td>8, 10, 12, 14</td>
<td>01--0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>11101</td>
<td>9, 13</td>
<td>01-01</td>
<td>8, 12, 9, 13</td>
<td>01-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>11110</td>
<td>10, 11</td>
<td>0101-</td>
<td>8, 12, 10, 14</td>
<td>01-0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**F = BCD'E + AB'C'D + B'C'E' + A'BC' + A'B'CD + BCD'E + A'E'**
6.12 (a) \[ f = A'C' + C'DE + A'B'D'E' + ACD'E + AB'CD + ABCE' \]


6.12 (b) \[ f = A'C' + C'DE + A'B'D'E' + AB'CE + ACDE' + ABCD' \]


Essential prime implicants: \( A'C'D'E', BDE, A'BCE, BCD \)


There are four minimal choices from the first parenthesis. In the second parenthesis only UZ is minimal since Z has fewer literals than the other two PI's. The minimal solutions are \((QW+T)(RX+U)(SY+V)(W+Y)(X+Z))\).
6.13 (b) \( f = BCD + A'B'CE + B'DE + A'C'D'E' + ABD + B'C'D'E + AB'D'E + B'C'D'E' + A'B'C'D' \)

\( f = BCD + A'B'CE + BDE + A'C'D'E' + ABD + B'C'D'E' + AB'D'E + B'DE + B'CD'E' + B'C'D'E \)

\( f = BCD + A'B'CE + BDE + A'C'D'E' + ABD + B'C'D'E' + AB'D'E + B'CD'E' + B'C'D'E \)

6.13 (b) \( f = BCD + A'B'CE + BDE + A'C'D'E' + ABD + B'C'D'E' + AB'D'E + B'DE + B'CD'E' + B'C'D'E \)

**Prime implicants of \( f' \):** \( AB'D'E', BCDE, AC'D', AC'E', A'CE, A'CD, A'BC \)

6.13 (a) \( f(A, B, C, D, E) = AB'D'E' + BCDE + AC'D' + AC'E' + A'CE + A'CD + A'BC \)


**Prime implicants of \( f' \):** \( A'B'C'D'E', AC'D'E', A'B'CE, BC'D'E', BCDE', B'DE, B'CD, ABD' \)

6.13 (b) \( f = A'B'C'D'E' + AC'D'E' + A'B'CE + BC'D'E' + BCDE' + B'DE + B'CD + ABD' \)

\( f = (A' + B' + D)(A' + C + D + E)(B' + C' + D + E')(B' + C' + D')(A + B + C' + E')(B + D' + E')(A + B' + C + D' + E) \)
Unit 6 Solutions

6.14 (a)

<table>
<thead>
<tr>
<th>( A'BCD )</th>
<th>( A'B'D' )</th>
<th>( ABC'D )</th>
<th>( A'D' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 3</td>
<td>00-1*</td>
<td>0-01*</td>
<td>1, 5, 9, 13</td>
</tr>
<tr>
<td>4</td>
<td>0100*</td>
<td>0-01</td>
<td>4, 5, 12, 13</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>0-01</td>
<td>4, 6, 12, 14</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>010-</td>
<td>8, 9, 12, 13</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>010-</td>
<td>8, 10, 12, 14</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>-100</td>
<td>12, 13, 14, 15</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>-100</td>
<td>-10-</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>100-</td>
<td>10-</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>100-</td>
<td>10-</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>-101</td>
<td>10-</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>-110</td>
<td>10-</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>111</td>
<td>10-</td>
</tr>
</tbody>
</table>

Prime Implicants: \( A'B'D', A B, A C', C'D, A D' \)

6.14 (b)

<table>
<thead>
<tr>
<th>( A'B'D' )</th>
<th>( AB'D' )</th>
<th>( A'B'D' )</th>
<th>( A'B'D' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0-00*</td>
<td>0-00*</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>0-00*</td>
<td>0-00*</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>0-00*</td>
<td>0-00*</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>010-</td>
<td>010-</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>101*</td>
<td>101*</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>101</td>
<td>101</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>111</td>
<td>111</td>
</tr>
</tbody>
</table>

Prime Implicants of \( f' \): \( A'B'CD, A'B'D', ABC'D, AB'C, B'CD', A'C'D' \)

6.15 (a)

<table>
<thead>
<tr>
<th>( A'B'D' )</th>
<th>( A'B'D' )</th>
<th>( A'B'D' )</th>
<th>( A'B'D' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00001</td>
<td>1, 5</td>
<td>00-01</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>1, 9</td>
<td>00-01</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>1, 17</td>
<td>00-01</td>
</tr>
<tr>
<td>5</td>
<td>0010</td>
<td>2, 6</td>
<td>00-01</td>
</tr>
<tr>
<td>6</td>
<td>0100</td>
<td>4, 5</td>
<td>00-01</td>
</tr>
<tr>
<td>9</td>
<td>0100</td>
<td>4, 5</td>
<td>00-01</td>
</tr>
<tr>
<td>12</td>
<td>0110</td>
<td>4, 12</td>
<td>010-</td>
</tr>
<tr>
<td>17</td>
<td>1001</td>
<td>4, 20</td>
<td>010-</td>
</tr>
<tr>
<td>20</td>
<td>1010</td>
<td>5, 7</td>
<td>001-</td>
</tr>
<tr>
<td>7</td>
<td>0011</td>
<td>5, 13</td>
<td>010-</td>
</tr>
<tr>
<td>13</td>
<td>0110</td>
<td>6, 7</td>
<td>010-</td>
</tr>
<tr>
<td>22</td>
<td>1010</td>
<td>6, 22</td>
<td>010-</td>
</tr>
<tr>
<td>25</td>
<td>1100</td>
<td>9, 13</td>
<td>010-</td>
</tr>
<tr>
<td>28</td>
<td>1100</td>
<td>9, 25</td>
<td>010-</td>
</tr>
<tr>
<td>15</td>
<td>0111</td>
<td>12, 13</td>
<td>010-</td>
</tr>
<tr>
<td>30</td>
<td>1110</td>
<td>12, 28</td>
<td>010-</td>
</tr>
<tr>
<td>17</td>
<td>25, 1001</td>
<td>001-</td>
<td></td>
</tr>
<tr>
<td>20, 22</td>
<td>101-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20, 28</td>
<td>101-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>101-</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>15</td>
<td>101-</td>
<td></td>
</tr>
<tr>
<td>22, 30</td>
<td>110-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28, 30</td>
<td>110-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Prime Implicants: \( a c e', a'c e, c'd'e', a'c d', a'b'c, b'c e', a'b'd e', c'd'e', a'd'e' \)

<table>
<thead>
<tr>
<th>( A'B'C' )</th>
<th>( A'B'C' )</th>
<th>( A'B'C' )</th>
<th>( A'B'C' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>25</td>
<td>28</td>
<td>30</td>
</tr>
</tbody>
</table>

Essential Prime Implicants: \( a c e', c'd'e', a'c e, a'b'd e' \)

\begin{align*}
f &= a c e' + c'd'e + a'c e + a'b'd e' + a'c d' \\
f &= a c e' + c'd'e + a'c e + a'b'd e' + c d'e' \\
\end{align*}

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
### Unit 6 Solutions

#### 6.15 (b)

<table>
<thead>
<tr>
<th></th>
<th>Prime Implicants of ( f' ): ace, ade, ac'd, ac'e', bc'd, a'bde', bc'e', c'de, c'd'e'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0, 8, 0-000(\checkmark)</td>
</tr>
<tr>
<td>8</td>
<td>01000(\checkmark)</td>
</tr>
<tr>
<td>16</td>
<td>10000(\checkmark)</td>
</tr>
<tr>
<td>3</td>
<td>00011(\checkmark)</td>
</tr>
<tr>
<td>10</td>
<td>01010(\checkmark)</td>
</tr>
<tr>
<td>18</td>
<td>10010(\checkmark)</td>
</tr>
<tr>
<td>24</td>
<td>11000(\checkmark)</td>
</tr>
<tr>
<td>11</td>
<td>01011(\checkmark)</td>
</tr>
<tr>
<td>14</td>
<td>01110(\checkmark)</td>
</tr>
<tr>
<td>19</td>
<td>10011(\checkmark)</td>
</tr>
<tr>
<td>21</td>
<td>10101(\checkmark)</td>
</tr>
<tr>
<td>26</td>
<td>11010(\checkmark)</td>
</tr>
<tr>
<td>23</td>
<td>10111(\checkmark)</td>
</tr>
<tr>
<td>27</td>
<td>11011(\checkmark)</td>
</tr>
<tr>
<td>29</td>
<td>11101(\checkmark)</td>
</tr>
<tr>
<td>31</td>
<td>11111(\checkmark)</td>
</tr>
</tbody>
</table>

#### 6.16

<table>
<thead>
<tr>
<th></th>
<th>Prime Implicants of ( f' ): ace, a'bde', c'de, c'd'e'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000001(\checkmark)</td>
</tr>
<tr>
<td>2</td>
<td>000010(\checkmark)</td>
</tr>
<tr>
<td>16</td>
<td>100000(\checkmark)</td>
</tr>
<tr>
<td>32</td>
<td>000000(\checkmark)</td>
</tr>
<tr>
<td>3</td>
<td>000011(\checkmark)</td>
</tr>
<tr>
<td>17</td>
<td>010001(\checkmark)</td>
</tr>
<tr>
<td>18</td>
<td>010010(\checkmark)</td>
</tr>
<tr>
<td>48</td>
<td>110000(\checkmark)</td>
</tr>
<tr>
<td>19</td>
<td>010011(\checkmark)</td>
</tr>
<tr>
<td>26</td>
<td>011010(\checkmark)</td>
</tr>
<tr>
<td>28</td>
<td>011100(\checkmark)</td>
</tr>
<tr>
<td>15</td>
<td>001111 A'B'C'DEF</td>
</tr>
<tr>
<td>29</td>
<td>011101(\checkmark)</td>
</tr>
<tr>
<td>30</td>
<td>011110(\checkmark)</td>
</tr>
<tr>
<td>39</td>
<td>100111 A'B'C'DEF</td>
</tr>
<tr>
<td>63</td>
<td>111111 A'B'C'DEF</td>
</tr>
</tbody>
</table>
6.16 (contd) 

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>26</th>
<th>32</th>
<th>39</th>
<th>48</th>
<th>63</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>A'B'CDEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>AB'CDEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>ABCDEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16,48</td>
<td>BC'D'E'F'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32,48</td>
<td>AC'D'E'F'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18,26</td>
<td>A'BD'EF'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26,30</td>
<td>A'BCEF'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28,29</td>
<td>A'BCDE'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28,30</td>
<td>A'BCDF'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,3,17,19</td>
<td>A'C'D'F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,3,18,19</td>
<td>A'C'D'E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16,17,18,19</td>
<td>A'BC'D'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.16 (a) \[ G = AB'C'DEF + ABCDEF + A'C'D'F + A'C'D'E + AC'D'E'F' + A'BC'D' + A'BCEF' \]
\[ G = AB'C'DEF + ABCDEF + A'C'D'F + A'C'D'E + AC'D'E'F' + A'BC'D' + A'BCEF' \]

6.16 (b) Essential prime implicants are underlined in 6.16 (a).

6.16 (c) If there were no don't cares, prime implicants 15, (26, 30), (28, 29), and (28, 30) are omitted. There is only one minimum solution. Same as (a), except delete the second equation.

6.17 (a) 

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>000001√</th>
<th>1, 33</th>
<th>-00001*</th>
<th>11, 15, 43, 47</th>
<th>-01-11*</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>001100*</td>
<td>33, 35</td>
<td>1000-1*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>100001√</td>
<td>7, 15</td>
<td>00-11*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>000111√</td>
<td>11, 15</td>
<td>001-11√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>001011√</td>
<td>11, 43</td>
<td>-01011√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>100011√</td>
<td>35, 43</td>
<td>10-011*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>110010√</td>
<td>50, 54</td>
<td>110-10*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>001111√</td>
<td>50, 58</td>
<td>11-010*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>011110*</td>
<td>15, 47</td>
<td>-01111√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>101011√</td>
<td>43, 47</td>
<td>101-11√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>110110√</td>
<td>43, 59</td>
<td>1-1011*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>111010√</td>
<td>58, 59</td>
<td>11101-√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>111100*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>101111√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>111011√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Prime Implicants: A'B'CDEF', A'BCDEF', ABCDE'F', B'C'D'E'F, AB'C'D'F, A'B'DEF, AB'D'EF, ABC'EF', ABD'EF', ACD'EF, ABC'D'E, B'CEF
Unit 6 Solutions

6.17 (a)  
(continued)

| Essential Prime Implicants: \( A'B'CDEF', ABCD'E', B'C'D'E, AB'D'E, B'C'E'F \)  
| \( G' = BC' + AC'D + A'B + EF' + A'C'D'E + AB'F' + BDF + C'D'E' + DE'F + C'F' \)  
| \( G' = BC' + AC'D + A'B + EF' + A'C'D'E + AB'F' + BDF + C'D'E' + DE'F + C'F' \)  
| \( G' = BC' + AC'D + A'B + EF' + A'C'D'E + AB'F' + BDF + C'D'E' + DE'F + D'F' \)  
| \( G' = BC' + AC'D + A'B + EF' + A'C'D'E + AB'F' + C'DE' + DE'F + C'F' + BDE \)  

6.17 (b)  
Prime Implicants of \( G' \): \( BC', AC'D, A'B, EF', A'C'D'E \)  

| Essential Prime Implicants: \( A'B'CDEF', ABCD'E', B'C'D'E, AB'D'E, B'C'E'F \)  
| \( G = A'B'CDEF' + ABCD'E' + B'C'D'E' + AB'D'E + A'B'EF + B'C'EF + ACD'E'F + AB'D'EF \)  
| \( G = A'B'CDEF' + ABCD'E' + B'C'D'E' + AB'D'E + A'B'EF + B'C'EF + ACD'E'F + AB'D'EF \)  
| \( G = A'B'CDEF' + ABCD'E' + B'C'D'E' + AB'D'E + A'B'EF + B'C'EF + ABCD'E + AB'C'D'F \)  
| \( G = A'B'CDEF' + ABCD'E' + B'C'D'E' + AB'D'E + A'B'EF + B'C'EF + ABCD'E + AB'D'EF \)  

Essential Prime Implicants: \( A'B'CDEF', ABCD'E', B'C'D'E, AB'D'E, B'C'E'F \)

6.18  
(a) \(-0-1 = (1, 3, 9, 11), -01- = (2, 3, 10, 11), --11 = (3, 7, 11, 15), 1--1 = (9, 11, 13, 15)\)  
(b) Maxterms = 0, 4, 5, 6, 8, 12, 14  
(c) Don't cares = 1, 10, 15  
(d) \( B'C, CD, AD \)

6.19  
Package

<table>
<thead>
<tr>
<th>Cart</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
</tr>
</tbody>
</table>

Using Petrick’s method:

\( (C1 + C3)(C2 + C3 + C5)(C1 + C4)(C1 + C5) \)  
\( (C2 + C3)(C2 + C3 + C4)(C3 + C4) \)  
\( = (C1C2 + C1C5 + C3)(C1 + C4C5)(C2C4 + C3) \)  
\( = (C1C2 + C1C5 + C1C3 + C3C4C5)(C2C4 + C3) \)  
\( = C1C2C4 + C1C3 + C3C4C5 \)

Each product term specifies a nonredundant combination of carts that can be used to deliver the packages. The minimal cart solution, using carts C1 and C3, costs $6. However, using the three carts C1, C2 and C4 costs only $5 so it is the minimal cost solution desired by the stockroom manager.
6.21 Prime implicants: $AC, AD', AB, CD, BD, A'D$

Minimum solutions: 
- $(AD' + CD)$
- $(AD' + BD)$
- $(AB + BD)$
- $(AB + CD)$
- $(AB + A'D)$

6.22 (a) 

\[
F = AC'D + BCD' + A'D' + E
\]

6.22 (b) 

\[
Z = CD' + ACD' + E(B'C' + B'D) + FCD' + G(A'C')
\]

6.23 (a) Each minterm of the four variables $A, B, C, D$ expands to two minterms of the five variables $A, B, C, D, E$. For example,

\[
m_4(A,B,C,D) = A'BC'D' = A'BC'D'E' + A'BC'D'E
\]

$G = C'E'F' + DEF + A(D'F') + B(DF)$

6.24 

* This square contains $1 + B$, which reduces to $1$. 

\[
G = C'E'F' + DEF + A(D'F') + B(DF)
\]
**Unit 7 Problem Solutions**

7.1 (a) 

\[ f = a'b'd' + a'b'c' + a'b + a'b'd' \]

Sum of products solution requires 5 gates, 16 inputs.

7.1 (b) 

Beginning with the minimum sum of products solution, we can get

\[ f = a'b (c + d') + ab' (c' + d') \]

5 gates, 12 inputs

So sum of products solution is minimum.

7.2 (a) 

\[ AC'D + ADE' + BE' + BC' + A'D'E' = E'(AD + B) + A'D'E' + C'(AD + B) \]

\[ F = (AD + B)(E' + C') + A'D'E' \]

4 levels, 6 gates, 13 inputs

7.2 (b) 

\[ AE + BDE + BCE + BCFG + BDFG + AFG = AE + AFG + BE(C + D) + BFG(C + D) \]

\[ F = (E + FG)[A + B(C + D)] \]

4 levels, 6 gates, 12 inputs
Unit 7 Solutions

7.3 \( F(a, b, c, d) = a'b'd + ac'd \) or \( d(a'b + ac') = d(a + b)(a' + c') \)

You can obtain this equation in the product of sums form using a Karnaugh map, as shown below:

\[
F = a'b'd + ac'd \\
(F)' = [(a'b'd + ac'd)]'
\]

\[
7.4 \quad F(A, B, C, D) = \sum m(5, 10, 11, 12, 13)
\]

\[
F = ABC' + BC'D + AB'C = BC'(A + D) + AB'C
\]

\[
F = BC' (A + D) + AB'C
\]

4 gates, 10 inputs
## 7.5

\[
Z = (A + C + D)(A' + D')(A' + C')(A' + B')
\]

## 7.6

\[
Z = ABC + AD + C'D' \\
= A(BC + D) + C'D'
\]

## 7.7

\[
Z = AE + BDE + BCEF \\
= E(A + BD + BCF) \\
= E[A + B(D + CF)]
\]

## 7.8

For the solution to 7.8, see FLD p. 700.

## 7.9

\[
f_1 = acd' + ad + a'b'd
\]

\[
f_2 = a'd' + a'b'd + acd'
\]

\[
f_3 = ab'c + b'cd + ac'd'
\]

\[
f_4 = ab'c + b'cd + b'cd' + ab'd'
\]

\[
f_5 = ab'c + b'cd + a'bd'
\]

\[
f_6 = a'b'd' + a'b'd + acd'
\]

## 7.10

\[
f_1(A, B, C, D) = \sum m(3, 4, 6, 9, 11)
\]

\[
f_2(A, B, C, D) = \sum m(2, 4, 8, 10, 11, 12)
\]

\[
f_3(A, B, C, D) = \sum m(3, 6, 7, 10, 11)
\]

\[
f_4 = ab'c + b'cd + a'bd'
\]

\[
f_5 = a'b'd' + a'b'd + acd'
\]

\[
f_6 = ab'c + b'cd + a'bd'
\]
7.11

F_1 = (a + c)(a + b')(a^b + c)(a^b + c')

\[ F_2 = (b'^c + d)(a'^b' + c')(a^b + c') \]
\[ F_2 = (a^b + d)(a'^b' + c')(a^b + c') \]

8 gates

7.12

f_1 = (A + B + C)(B' + D)

f_2 = (A + B + C)(B' + C + D)(A' + C)

f_3 = (B' + C + D)(A + C)(B' + C)

9 gates

7.13 (a) Using F = (F')' from Equations (7.23(b)), p. 206:

\[ f_1 = [(A'B'D)'(ABD)'(AB'C)'(B'C)'] \]
\[ f_2 = [C'(A'BD)'] \]
\[ f_3 = [(BC)'(AB'C)'(ABD)'] \]

7.13 (b) Using F = (F')' from Equations derived in problem 7.12:

\[ f_1 = [(A + B + C)' + (B' + D)']' \]
\[ f_2 = [(A + B + C)' + (B' + C + D)' + (A' + C)']' \]
\[ f_3 = [(B' + C + D)' + (A + C)' + (B + C)']' \]
7.14 (a)  
\[ f = (a + b + c) (a + b + d') (a' + b' + d') (a' + b' + c') \]
5 gates, 16 inputs

and 
\[ f = a'b + ab' + b'cd' + ac'd' \]
\[ f = a'b + ab' + a'cd' + bc'd' \]
(two other minimum solutions)
5 gates, 14 inputs minimal

7.14 (b)  
Beginning with the sum of products solution, we get
\[ f = a'b + ab' + d' (a'c + ac') \]
\[ f = a'b + ab' + d' (a' + c') (a + c) \] — 6 gates, 14 inputs

But, beginning with the product of sums solution above, we get
\[ f = (a + b + cd') (a' + b' + c'd') \] — 5 gates, 12 inputs, which is minimum

7.15 (a)  
From K-maps:
\[ F = a'c + bc'd + ac'd \] — 4 gates, 11 inputs
\[ F = (a + b + c) (c + d) (a' + c') \] — 4 gates, 10 inputs, minimal

7.15 (b)  
From K-maps:
\[ F = cd + ac + b'c' \] — 4 gates, 9 inputs
\[ F = (b' + c) (a + c' + d) \] — 3 gates, 7 inputs, minimal

7.15 (c)  
From K-maps:
\[ F = ad + a'cd' + bcd \]
\[ = ad + a'cd' + a'bc \] — 4 gates, 11 inputs
\[ F = (a + c) (a' + d) (a + b + d') \] — 4 gates, 10 inputs, minimal

7.15 (d)  
From K-maps:
\[ F = a'b + ac + bd' \] — 4 gates, 9 inputs, minimal
\[ F = (a + b) (a' + c + d') (a' + b + c) \]
\[ = (a + b) (a' + c + d') (b + c + d) \] — 4 gates, 11 inputs

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Unit 7 Solutions

7.16 (a) In this case, multi-level circuits do not improve the solution. From K-maps:

\[ F = ABC' + ACD + A'BC + A'C'D \] — 5 gates, 16 inputs, minimal

\[ F = (A' + B + C)(A + C + D)(A' + C' + D) \]
\[ (A + B + C') \] — 5 gates, 16 inputs, also minimal

Either answer is correct.

7.16 (b) Too many variables to use a K-map; use algebra. Add \( ACE \) by consensus, then use \( X + XY = X \)

\[ ABCE + ABEF + ACD' + ABEG + ACDE + ACE \]
\[ = ABEF + ACD' + ABEG + ACDE + ACE \]
\[ = F = ABE (F + G) + AC (D' + E) \]

5 gates, 13 inputs, minimal

7.17 (a) Too many variables to use a K-map; use algebra. Add \( ACE \) by consensus, then use \( X + XY = X \)

\[ \prod M(0, 1, 2, 4, 8) \]

7.17 (b) \[ F = (A + C + D)(A + B + C) \]
\[ (A + B + D)(B + C + D) \]
\[ = (A + D + BC)(B + C + AD) \] or
\[ = (A + C + BD)(B + D + AC) \] or
\[ = (C + D + AB)(A + B + CD) \]

This solution has 5 gates, 12 inputs. Beginning with the sum of products requires 6 gates.

\[ F = \prod M(0, 1, 2, 4, 8) \]
7.18 (a) \( F(w, x, y, z) = (x + y' + z)(x' + y + z) \)

- **OR-AND**
- **NOR-NOR**
- **AND-NOR**
- **NAND-AND**

From Karnaugh map: \( F = wxy + wx'y' + wz \)

7.18 (b) \( F(a, b, c, d) = \sum m(4, 5, 8, 9, 13) \)

From Kmap:

- \( F = a'bc' + ab'c' + bc'd \)
- \( F = a'bc' + ab'c' + ac'd \)

\( F = c'(a + b) \left( a' + b' + d \right) \)
Unit 7 Solutions

7.19 (a) 

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>f = (y' + z)(x' + y + z')</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

7.19 (b) 

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>f = yz + y'z' + x'y'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

From Kmap: 
\[
F = (y' + z)(x' + y + z')
\]

7.20 (a) Using OR and NOR gates:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>f = a'b + cd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

7.20 (b) Using NOR gates only:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>f = (b + c)(b + d')(a' + c)(a' + d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

7.21 (a) NAND gates: 
\[
F = D' + B'C + A'B
\]

NOR gates: 
\[
F = (A' + B' + D')(B + C + D')
\]

7.21 (b) NAND gates: 
\[
f = a'b'c' + ac'd' + b'cd
\]

NOR gates: 
\[
f = (b' + c')(c' + d)(a + b + c)(a' + c + d')
\]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
7.21 (c) NAND gates:
\[ f = a'b'd' + bc'd + cd' \]
NOR gates:
\[ f = (b + d')(b' + d)(a' + c)(b' + c') \]
\[ f = (b + d')(b' + d)(a' + c)(c' + d') \]

7.21 (e) NAND gates:
\[ F = A'C'D + A'B'E + C'D + A'B'D + B'D'E + A'B'D'E' \]
\[ F = A'C'D + A'B'E + C'D + A'B'D + B'D'E + A'B'C'E + B'D'E \]
\[ F = A'C'D + A'B'E + C'D + A'B'D'E + A'B'C'E + B'D'E \]
NOR gates:
\[ F = (A + C' + D + E)(C + D' + E)(A + B' + E) \]
\[ (A' + B + D' + E)(A' + B + C)(B' + D + E') \]

7.21 (f) NAND gates:
\[ f = c'd' + a'b + a'd' + ab'c' \]
NOR gates:
\[ f = (a + b + d')(a' + b' + d')(a' + c) \]

7.21 (g) NAND gates:
\[ f = x'y' + wy + w'z' + wz \]
\[ f = x'y' + wy + wx' + w'z' \]
\[ f = x'y' + wy' + y'z' + wz \]
NOR gates:
\[ f = (w + x' + z')w + y'z + wz \]
\[ f = (w + x' + z')w + y'z + wz \]
\[ f = (w + x' + z')w + y'z + wz \]

7.22 (a) F is 0 if any 3 (or 4) of the inputs are 1 so
\[ F = (A + B' + C' + D')(A' + B' + C + D) \]
\[ (A' + B' + C' + D)(A + B + C' + D) \]
\[ (A' + B' + C' + D') \]
\[ (A' + B' + C' + D') \]

(b) \[ F = (A' + B' + C'D')(A'B' + C' + D') \] or
\[ F = (A' + C' + B'D')(A'C' + B' + D') \]
Unit 7 Solutions

7.23 (b)  

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ f' = (a + b')(c' + d') \]

\[ f = A(B' + C') + A'BC + B'C' \]

7.23 (c)  

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ f = a'b + cd \]

7.24 (a)  

7.24 (b)  

\[ f = A(AB)' + (AB)[AB + B' + C]B + [AB + B' + C]C' \]
\[ = AB' + (A' + B'[AB + BC] + AC' + B'C' \]
\[ = AB' + A'BC + AC' + B'C' \]

7.24 (c)  

\[ f = A(B' + C') + A'BC + B'C' \]

7.25 (a)  

7.25 (b)  

\[ f = A(B' + C') + A'BC + B'C' \]
\[ = [A + A'BC + B'C']B' + C' + A'BC + B'C' \]
\[ = [A + BC + B'C'][A' + B' + C'] \]
\[ = [A + B + B'C'][A + C + B']B' + C' \]
\[ = [A + B + C'][A + C + B'][A' + B' + C'] \]

7.25 (c)  

\[ f = (A + B + C')\vphantom{\prod}(A + C + B')(A' + B' + C') \]
\[ = [A + (B + C')(B' + C')(A' + B' + C')] \]
\[ = [A + BC + B'C']A' + B' + C' \]

74

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
7.26

7.27 (a)

7.27 (b) \( f = (A' + B)(C' + CD)(D' + CD) + B'(C' + D')C \\
= (A' + B)(C' + D)(D' + C) + B'CD' \\
= (A' + B)(C'D' + CD) + B'CD' \\
= A'C'D' + A'CD + BC'D' + BCD + B'CD' \\

7.27 (c) Remove the inverter from the output NAND so the new output is \( f' \). Then, ‘push the inverters at the NAND outputs forward to the inputs of the following gates’. An AND with inverters on the inputs is a NOR

7.28 (a)

7.28 (b) \( f = (b' + d' + ae) (b + c' + de') (a + c + d) \)

7.29

\[ f = (a' + d) (a' + b + c) (a + b') \]
\[ = (a + b') [a' + d (b + c)] \]
\[ = (a + b') (a' + bd + cd) \]
Unit 7 Solutions

7.30 (a) \( Z = abe'f + c'e'f + d'e'f + gh \)
\[ = ef(ab + c' + d') + gh \]

7.30 (b) \( Z = (a' + b + e + f)(c' + a' + b)(d' + a' + b)(g + h) \)
\[ = [a' + b + c'd' (e + f)] (g + h) \]

7.31 \( F = abde' + a'b' + c \)
\[ = (a + b') (a' + bde') + c \]
\[ = (a + b' + c) (a' + c + bde') \]
Alternate: \( F = (a' + b + c) (b' + c + ade') \)

7.32 \( f = x'yz + xy'w' + xvy'z' \)
\[ = x'yz + xvy'(z' + w') \]

7.33 (a)
\[
\begin{array}{|c|c|c|c|}
\hline
C & D & A & B \\
\hline
00 & & & \\
01 & & & \\
11 & & & \\
10 & & & \\
\hline
\end{array}
\]
\[ F = BCD + B'CD + A'B'D' + A'CD \]
\[ F = BCD + B'CD + A'B'D' + A'BD \]

Draw AND-OR circuit and replace all gates with NANDs.

7.33 (b)
\[
\begin{array}{|c|c|c|c|}
\hline
C & D & A & B \\
\hline
00 & & & \\
01 & & & \\
11 & & & \\
10 & & & \\
\hline
\end{array}
\]
\[ F = (B + C + D')(B' + D)(A' + D)(A' + B' + C) \]

Draw OR-AND circuit and replace all gates with NORs.

7.33 (c) \( F = B (A'D + C'D) + B' (A'D' + CD) \)

Alternative:
\[
F = A' (B'D' + BD) + D (B'C + BC')
\]
\[ = D (A'B + BC') + B' (A'D' + CD) \]
\[ = A' (B'D' + CD) + D (B'C + BC') \]
\[ = D (A'C + BC') + B' (A'D' + CD) \]
7.34 (a) \[
F = ABCD + AB'C'D' + A'B'CD + A'B'CD' + A'B'C'D'
\]

7.34 (b) \[
F = (A + C + D')(B + C' + D) (A + B' + C)
\]

7.34 (c) Many solutions exist. Here is one, drawn with alternate gate symbols.
\[
F = A'(B'C'D' + B'CD + BCD') + A (B'C'D + BC'D' + BCD)
\]
\[
= A'(B'(C'D + CD) + BCD') + A (B(C'D' + CD) + B'C'D)
\]

7.35 (a) \[
F = A'BC' + BD + AC + B'CD'
\]
\[
= B (D + A'C') + C (A + B'D')
\]

Many NOR solutions exist. Here is one.
\[
F = (B + C) (A' + C + D) (A + B + D')(A + B' + C' + D)
\]
\[
= (B + C) [A + (B + D')(B' + C' + D)] (A' + C + D)
\]
\[
= (B + C) [A (C + D) + A'(B + D') (B' + C' + D)]
\]
\[
= (B + C) [A (C + D) + A'(B(C' + D) + B'D')]
\]
Unit 7 Solutions

7.35 (b)

\[
F = (A + C)(B + D)(A' + B + C')(B' + C + D')
\]

7.36

\[
F = \sum m(0, 1, 2, 3, 4, 5, 7, 9, 11, 13, 14, 15)
F = D + A'B' + A'C' + ABC
= D + A' (B' + C') + ABC
\]

Alternate solution:
\[
F = D + (A' + BC) (A + B' + C')
\]

7.36 (a)

7.36 (b)
7.36 (c)

\[ Z = A \left[ BC' + D + E(F' + GH) \right] \]

7.37

7.38

(a) No—NOR-AND is equivalent to NOT-AND-AND.
(b) Yes—NOR-OR is equivalent to OR-AND-NOT.
(c) No—NOR-NAND is equivalent to OR-OR.
(d) Yes—NOR-XOR is equivalent to NOT-AND-XOR.
(e) Yes—NAND-AND is equivalent to NOT-OR-AND.
(f) No—NAND-OR is equivalent to NOT-OR-OR.
(g) No—NAND-NOR is equivalent to AND-AND.
(h) Yes—NAND-XOR is equivalent to AND-XOR or AND-XOR-NOT.

7.39

\[
\begin{array}{|c|c|c|c|}
\hline
f_1 & a & b & c & d \\
\hline
& 00 & 01 & 11 & 10 \\
\hline
\hline
00 & 1 & 1 & X \\
\hline
11 & 1 & 1 & X \\
\hline
10 & X & X & X \\
\hline
\end{array}
\]

\[ f_1 = b c' d' + a c \]

\[
\begin{array}{|c|c|c|c|}
\hline
f_2 & a & b & c & d \\
\hline
& 00 & 01 & 11 & 10 \\
\hline
\hline
00 & 1 & 1 & 1 & 1 \\
\hline
01 & 1 & 1 & 1 & 1 \\
\hline
11 & X & X & X & X \\
\hline
10 & X & X & X & X \\
\hline
\end{array}
\]

\[ f_2 = b c' + b' c' d' \]

8 gates

\[
\begin{array}{|c|c|c|c|}
\hline
f_3 & a & b & c & d \\
\hline
& 00 & 01 & 11 & 10 \\
\hline
\hline
00 & 1 & 1 & 1 & 1 \\
\hline
01 & 1 & 1 & 1 & 1 \\
\hline
11 & X & X & X & X \\
\hline
10 & X & X & X & X \\
\hline
\end{array}
\]

\[ f_3 = a b + a d + b c d' \]
Unit 7 Solutions

7.40

7.41

7.42 (a)

7.42 (b)

Circle 1's to get sum-of-products expressions:

\[ f_1 = x'yz + x'yz' + xy' \]

\[ f_2 = y'z + x'yz + xyz' \]

8 gates

\[ f_3 = xy' + y'z + x'yz' + xyz' \]

\[ f_1 = a'c + a'b'd + a'b'd' \]

\[ f_2 = a'b' + a'b'd + a'b'd' \]

6 gates

\[ f_3 = \overline{a'b' + a'b'd + a'b'd'} \]

6 gates

\[ f_1 = \overline{a'c + a'b'd + a'b'd'} \]

8 gates

\[ f_2 = \overline{a'b' + a'b'd + a'b'd'} \]

6 gates

\[ f_3 = \overline{a'c + a'b'd + a'b'd'} \]

6 gates

\[ f_1 = (a' + b + d) (b' + c' + d') (b' + d) \]

\[ f_2 = (a' + b + d) (b' + c' + d') (b + d') \]

6 gates

\[ f_1 = (a' + b + d) (b' + c' + d') (b' + d) \]

6 gates

\[ f_2 = (a' + b + d) (b' + c' + d') (b + d') \]

6 gates

\[ f_1 = \overline{(a' + b + d) (b' + c' + d') (b' + d)} \]

6 gates

\[ f_2 = \overline{(a' + b + d) (b' + c' + d') (b + d')} \]

6 gates

\[ f_1 = \overline{(a' + b + d) (b' + c' + d') (b' + d)} \]

6 gates

\[ f_2 = \overline{(a' + b + d) (b' + c' + d') (b + d')} \]

6 gates

Then convert directly to NAND gates.
7.43 (a) Circle 0's

\[ f_1 = (a + c + d) (b' + c') (c' + d) \]

7 gates

7.43 (b) Circle 1's to get sum-of-products expressions:

\[ f_1 = a'c' + c'd + b'cd \]

7 gates

7.44 (a)

\[ f_1 = (b + d) (c + d) (b + c) (a + c + d') \]

7.44 (b)

\[ f_1 = (b + d) (c + d) (b + c) (a + c + d') \]

\[ f_2 = b'd' + c'd + a b c'd \]
7.45 (a)  

\[ f_1 = a'd' + a'bc + acd' \]

\[ f_2 = a'c' + a'bc + acd' \]

7.45 (b)  

\[ f_1 = (a' + c)(a' + d')(c + d')(b + c' + d') \]

\[ f_2 = (a' + c)(a' + d')(b + c' + d')(a + c' + d) \]

7.46 (a)  

The circuit consisting of levels 2, 3, and 4 has OR gate outputs. Convert this circuit to NAND gates in the usual way, leaving the AND gates at level 1 unchanged. The result is:

7.46 (b)  

One solution would be to replace the two AND gates in (a) with NAND gates, and then add inverters at the output. However, the following solution avoids adding inverters at the outputs:

\[ F_1 = [(a + b')(c + d)(e' + f)] \]

\[ = ace' + b'ce' + de' + acf + b'cf + df \]

\[ = ce'(a + b') + d(e' + f) + cf(a + b') \]

\[ F_2 = [(a + b')c + g'](e' + f)h \]

\[ = h(ace' + b'ce' + acf + b'cf) + g'h(e' + f) \]

\[ = h[ce'(a + b') + cf(a + b')] + g'h(e' + f) \]
Unit 8 Problem Solutions

8.1

8.2 (a)

8.2 (a) (contd)

Static 1-hazards: 1101 ↔ 1111 and 0100 ↔ 0101

F = AC'D' + AC + BC'D

8.2 (b)

F = AC'D' + AC + BCD + A'BC + ABD

8.2 (c)

F' = (A + C')(A' + C + D)(B + C + D')

8.3 (a)

G = A'C'D + BC + A'BD

G = A'C'D + BC + A'BD

8.3 (b) Modified circuit (to avoid hazards)

8.4

A = 1; B = Z; C = 1 · Z = X; D = 1 + Z = 1;
E = X' = X; F = 1' = 0; G = X · 0 = 0;
H = X + 0 = X

See FLD Table 8-1, p. 231.
Unit 8 Solutions

8.5 \[ A = B = 0, \ C = D = 1 \]
So \[ F = AB'D + BC'D' + BCD = 0 \]

But in the figure, gate 4 outputs \( F = 1 \), indicating something is wrong. For the last NAND gate, \( F = 0 \) only when all its inputs are 1. But the output of gate 3 is 0. Therefore, gate 4 is working properly, but gate 3 is connected incorrectly or is malfunctioning.

8.6 (a)

\[ \begin{array}{c|cccccccc}
\text{G} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\text{F} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\text{H} & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\text{t (ns)} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array} \]

Glitch (static '0' hazard)

The circuit has three static 0-hazards:
0001 ↔ 0011, 1001 ↔ 1011 and 1000 ↔ 1010. Two sum terms are needed to eliminate the hazards:
\((A' + B)(B + D')\)

8.6 (b)

The minimal POS expression for \( f \) is \( f(a,b,c,d) = (a + d')(b' + d)(c' + d') \) but \((a + b')\) and \((b' + c')\) must be added to eliminate the static-0 hazards.

8.7 (a)

\[ f = (a+d')(b'+c+d)(a'+c'+d')(b'+c'+d) \]

The static-0 hazards are 0100 ↔ 0101, 0100 ↔ 0110, 0111 ↔ 0110, 1100 ↔ 1110, 1111 ↔ 1110, 0011 ↔ 1011 and 0111 ↔ 1111.

8.7 (b)

\[ \begin{array}{c|cccc}
\text{a} & \text{d'} & \text{d} & \text{b'} & \text{a} \\
\text{b'} & \text{d} & \text{b'} & \text{c'} & \text{a} \\
\text{c'} & \text{d'} & \text{b'} & \text{c} & \text{a} \\
\text{d} & \text{b} & \text{c} & \text{a} & \text{d'} \\
\text{a} & \text{d} & \text{b} & \text{c} & \text{a} \\
\text{d} & \text{b} & \text{c} & \text{a} & \text{d'} \\
\text{b} & \text{c} & \text{a} & \text{d} & \text{b'} \\
\text{c} & \text{a} & \text{d} & \text{b} & \text{c'} \\
\end{array} \]

8.8 (a)

\[ F = B'D + A'C + A'B'C + B'CD' . \]

Static-1 Hazards: 0000 ↔ 0010, 1101 ↔ 1001

84

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
8.8 (a) contd

\[ F = B \overline{D} + A'C + A \overline{B'}C' + A'B'D' \]

Static-1 Hazards: 0000 ↔ 1000, 1101 ↔ 1001

\[ F = B \overline{D} + A'C + A \overline{CD} + \overline{B'C'D'} \]

Static-1 Hazards: 0000 ↔ 0010, 1000 ↔ 1001

Hazard-free AND-OR circuit function:

\[ f(A, B, C, D) = BD + A'C + AC'D + B'C'D' + A'B'D' + AB'C' \]

8.8 (b)

\[ F = (A + B + C + D)(B' + C + D) \]

\[ (A' + B + C')(A' + B' + D) \]

Static-0 Hazard: 1110 ↔ 1010
Unit 8 Solutions

8.8 (b) contd

\[ F = (A + B + C + D')(B' + C + D) \]
\[ (A' + B + C')(A' + C + D) \]

Static-0 Hazard: 1100 ↔ 1110

Hazard-free OR-AND circuit function:
\[ f(A, B, C, D) = (A + B + C + D')(B' + C + D) \]
\[ (A' + B + C')(A' + B' + D)(A' + C' + D) \]

8.9 (a)

\[ f = AC' + A'B'D \]

8.9 (b) Since a circuit with NOR gates is desired, start with POS expressions for \( f \) that corresponds to a hazard-free OR-AND (NOR-NOR) circuit. From the Kar

\[ f = (A'B' + AC')(A + D) \]
\[ (A'B'D + C')(A + B'D) \]

When this expression is expanded to a POS, it does not contain any sum of the form \((X + X' + \beta)\) so the corresponding circuit is free of hazards. The three level NOR circuit is.

\[ f = (A'B'D + C')(A + B'D) \]

It is possible to start with a SOP that is free of hazards, namely, \( f = AC' + A'B'D + B'C'D \), and then factor it, e.g., the same result as above is obtained by \( f = (A + B'D)C + A'B'D = (A'B'D + C')(A + B'D) \).
8.10

<table>
<thead>
<tr>
<th>W</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Y</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>V</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Z</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

8.11 (a) contd

\[ f = A'BC + AB'D + B'D' \]

8.11 (a) From the Karnaugh map for \( f \), it is seen that a hazard-free POS expression for \( f \) requires all prime implicants.

\[
f = (A + B)(B' + D')(B + C')(C' + D')(A + D')
\]

\( f \) can be multiplied out as

\[
f = (A + B)(B' + D')(B + C')(C' + D')(A + D') = (AC' + B)(AB'C' + D')
\]

8.12

8.11 (a)

\[
\]

From the Karnaugh map and the \( BB'C' \) term

- static-1 hazard: 1100 ↔ 1000
- static-0 hazard: 0001 ↔ 0101
- potential dynamic hazards: 0000 ↔ 0100 and 1101 ↔ 1001

The circuit shows that only 0000 ↔ 0100 propagates over three paths.

8.13

Static 1-hazards lie between 1000 ↔ 1010 and 0010 ↔ 0011

Without hazards: \( Z' = AC'D' + A'CD + B'CD' + A'B'C + AB'D' \)
Unit 8 Solutions

8.14 \( A = Z; B = 0; C = Z' = X; D = Z \cdot 0 = 0; \)
\( E = Z; F = 0 + 0 + X = X; G = (0 \cdot Z)' = 0' = 1; \)
\( H = (X + 1)' = 1' = 0 \)

8.15 \( A = B = C = 1, \) so \( F = (A + B' + C' ) (A' + B + C') \)
\( (A' + B' + C) = 1 \)
But, in the figure, gate 4 outputs \( F = 0, \) indicating something is wrong. For the last NOR gate, \( F = 1 \) only when all its inputs are 0. But the output of gate 1 is 1. Therefore, gate 4 is working properly, but gate 1 is connected incorrectly or is malfunctioning.

8.16 (a) \( F (A, B, C, D) = \sum m(0, 2, 5, 6, 7, 9, 12, 13, 15) \)
There are 3 different minimum AND-OR solutions to this problem. The problem asks for any two of these.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>F = BD + AC' + A'B'D' + A'C D'</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01</td>
<td>11</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Solution 1: 1-hazards are between 0000 ↔ 0010 and 0111 ↔ 0110

Solution 2: 1-hazards are between 0010 ↔ 0110 and 0000 ↔ 0010

Solution 3: 1-hazards are between 0111 ↔ 0110 and 0000 ↔ 1000

Without hazards:
\[ F = (A + B + D') (A + B' + C + D) (A' + C' + D) (A' + B' + C') \]

8.16 (b)
\[ F = (A + B + D') (A + B' + C + D) (A' + C' + D) \]
\( 0 \)-hazard is between 1011 ↔ 0011

Either way, without hazard:
\[ F' = (A + B + D') (A + B' + C + D) (A' + C' + D) \]
\( (B + C' + D') (A' + B + C') \)
9.1 See FLD p. 703 for solution.

9.2 See FLD p. 703 for solution.

9.3 See FLD p. 704 for solution.

9.4 See FLD p. 704 and Figure 4-4 on FLD p.105.

9.5

<table>
<thead>
<tr>
<th>y0, y1, y2, y3</th>
<th>a b c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>X 1 0 0</td>
<td>0 1 1</td>
</tr>
<tr>
<td>XX 1 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>XXX 1</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

\[ a = y_3 + y_2 \]

\[ b = y_3 + y_1 y \]

\[ c = y_3 + y_2 + y_1 + y_0 \]

9.6 See FLD p. 705 for solution.

9.7 See FLD p. 705 for solution.

9.8 See FLD p. 705-706 for solution.

9.9 The equations derived from Table 4-6 on FLD p. 107 are:

\[ D = x'y'b + xy'b + xyb + y'b \]

\[ bout = x'b + x'y + yb \]

See p. p. 706 for PAL diagram.

9.10 Note: \( A_8 = A_4' \) and \( A_3 = A_4 \). Equations for \( A_4 \) through \( A_6 \) can be found using Karnaugh maps. See FLD p. 707-708 for answers.

9.11 (a) \[ F = C'D' + BC + AC \] \( \Rightarrow \) Use 3 AND gates

\[ F' = [C'D' + BC + AC]' = [(C + B + D) (A + C)]' = B'C'D + AC \] \( \Rightarrow \) Use 2 AND gates

9.11 (b) \[ F = A'B' + C'D' \] \( \Rightarrow \) Use 2 AND gates

\[ F' = (A'B' + C'D')' = (A + B)(C + D) = AC + AD + BC + BD \] \( \Rightarrow \) Use 4 AND gates

9.12 (a) See FLD p. 708, use the answer for 9.12 (b), but leave off all connections to 1 and 1'.

9.12 (b) See FLD p. 708 for solution.

9.13 Using Shannon’s expansion theorem:

\[ F = ab'c'd'e + bc'd'e + a'c'd'e + ac'd'e \]

= \( b'(ac'd'e + a'c'd'e + ac'd'e) + b (c'd'e + a'c'd'e + ac'd'e) \)

= \( b'(ade' + c + c') + a'c'd'e + b [(c' + a')d'e + ac'd'e] \)

= \( b'(ade' + a'd'e) + b (c'd'e + a'd'e + ac'd'e) \)

The same result can be obtained by splitting a Karnaugh map, as shown to the right.
Unit 9 Solutions

9.14 (a) \[ R = ab'h' + bch' + eg'h + fgh \]
\[ = (ab' + bch')h' + (eg' + fg)h \]
\[ = [(a)h' + (c)b]h' + [(e)g' + (f)g]h \]

9.14 (b)

There are many solutions. For example:

9.16

9.16 contd
9.18 Since the decoder outputs are negative, NAND gates are required. The excess-3 outputs are \( \Sigma m(5,6,7,8,9), \Sigma m(1,2,3,4,9), \Sigma m(0,3,4,7,8), \) and \( \Sigma m(0,2,4,6,8) \) so four 5-input NAND gates are needed with inputs corresponding to the minterms of the excess-3 outputs.

9.19 Using \( S1 = w \) and \( S0 = z \), \( I0 = x \), \( I1 = 1 \), \( I2 = y \) and \( I3 = 0 \) which does not require any gates.

Other answers: Using \( S1 = w \) and \( S0 = y \), \( I0 = x \), \( I1 = z \), \( I2 = 0 \) and \( I3 = z' \) which requires one inverter. Using \( S1 = w \) and \( S0 = x \), \( I0 = z \), \( I1 = 1 \), \( I2 = yz' \) and \( I3 = y \) which requires one inverter and one AND gate.

9.20 \( f(a, b, c, d, e) = a'b'c'd'e' + a'b'c'd'e + a'bc'd'e' + a'bc'd'e + a'bc'd'e + a'bc'd'e' + a'b'c'd'e' + ab'c'd'e' + ab'c'd'e' + ab'c'd'e' + ab'c'd'e' + ab'c'd'e + abc'd'e + abc'd'e + abc'd'e' 

= a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') + a(b'c'd'e') 

I0 = a, I1 = a, I2 = a, I3 = 0, I4 = a, I5 = a, I6 = a', I7 = 1, I8 = 0, I9 = 1, I10 = 0, I11 = a', I12 = 1, I13 = a', I14 = 0, I15 = 0
Unit 9 Solutions

9.21 (a) 
\[
\begin{array}{c|ccc|c}
\hline
x & y & c_{in} & Sum & C_{out} \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
\hline
\end{array}
\]

9.21 (b) 
\[
\text{C}_{in} \rightarrow \text{Sum} \rightarrow \text{C}_{out}
\]

9.22 (a) 
\[
\begin{array}{c|ccc|c}
\hline
x & y & b_{in} & Diff & B_{out} \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
\hline
\end{array}
\]

9.22 (b) 
\[
\text{B}_{in} \rightarrow \text{Diff} \rightarrow \text{B}_{out}
\]

9.22 (c) 
\[
\text{X} \rightarrow \text{X}' \rightarrow \text{Diff} \rightarrow \text{B}_{out}
\]
9.23 For a positive number \( A \), \(|A| = A\) and for a negative number \( A \), \(|A| = -A\). Therefore, if the number is negative, that is \( A[3] \) is 1, then the output should be the 2’s complement (that is, invert and add 1) of the input \( A \).

9.24

9.25

9.26 (a)

9.26 (b)
If any of the inputs $y_i$ through $y_7$ is 1, then $d$ of the 8-to-3 decoder should be 1. But in that case, $c_1$ or $c_2$ of one of the 4-to-2 decoders will be 1. So $d = c_1 + c_2$.

If one of the inputs $y_4, y_5, y_6,$ and $y_7$ is 1, then $a$ should be 1, and $b$ and $c$ should correspond to $a_2$ and $b_2$, respectively. Otherwise, $a$ should be 0, and $b$ and $c$ should correspond to $a_1$ and $b_1$, respectively. So $a = c_2, b = c_2a_2 + c_1a_1$, and $c = c_2b_2 + c_1b_1$.
9.29 (b) (contd)

\[ Y = S'T'U + STU + RU' \]

\[ Z = R'S'T'U + STU' + STU \]

9.29 (c)

<table>
<thead>
<tr>
<th>RS T U</th>
<th>VWYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>0 1 0 1</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>1 0 0 1</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 1 0 1</td>
</tr>
</tbody>
</table>

9.30 (a)

<table>
<thead>
<tr>
<th>RS T U</th>
<th>VWYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>0 1 0 1</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>1 0 0 1</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>1 0 1 1</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
</tr>
</tbody>
</table>
9.30 (c) 

<table>
<thead>
<tr>
<th>RSTU</th>
<th>VWYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 - -</td>
<td>1000</td>
</tr>
<tr>
<td>-1 - -</td>
<td>0100</td>
</tr>
<tr>
<td>0 - 10</td>
<td>0100</td>
</tr>
<tr>
<td>1 - 00</td>
<td>0100</td>
</tr>
<tr>
<td>11 - -</td>
<td>0010</td>
</tr>
<tr>
<td>0 - 11</td>
<td>0010</td>
</tr>
<tr>
<td>1 - 01</td>
<td>0010</td>
</tr>
<tr>
<td>01 - -</td>
<td>0001</td>
</tr>
<tr>
<td>0 - 01</td>
<td>0001</td>
</tr>
<tr>
<td>1 - 11</td>
<td>0001</td>
</tr>
</tbody>
</table>

9.30 (b) 

<table>
<thead>
<tr>
<th>T</th>
<th>U</th>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

V = RS'

W = R'TU' + R'TU' + S

<table>
<thead>
<tr>
<th>T</th>
<th>U</th>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Y = R'TU + R'TU + RS

Z = RTU + RS + R'TU

9.31 (a) 

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

F₁ = (A + B + C + D)(A' + C + D')(C' + D')

9.31 (a) (contd) 

F₂ = (A + B + C + D)(A' + C + D')(A + D')

Alternate solution:

F₁ = (a + b + c + d)(a + c' + d')(a' + d')

F₂ = (a + b + c + d)(a + b' + d')(c + d')
### 9.31 (b)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>( F_1 F_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>((cd'))</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>1 1</td>
</tr>
<tr>
<td>((bd'))</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>0</td>
<td>1 1</td>
</tr>
<tr>
<td>((ad'))</td>
<td>1</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>1 1</td>
</tr>
<tr>
<td>((ac))</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>0 1</td>
</tr>
<tr>
<td>((a'c'd))</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1 0</td>
</tr>
</tbody>
</table>

### 9.32 (a)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

### Equations

\[
W = A'D + C + B + AD' \\
X = A'C'D' + C D + A D + B C \\
Y = AD' + BD + A'C'D' \\
Z = AD + BC + BD' \\
\text{Alt: } Z = A + BC + BD' \\
\]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
9.32 (b)

<table>
<thead>
<tr>
<th>a b c d</th>
<th>WX Y Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 -1</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>-1 -1</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>0 0 -1</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>1 -0 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>0 0 -1</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>-1 -1</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>1 -1 1</td>
<td>0 1 0 1</td>
</tr>
<tr>
<td>-1 -1</td>
<td>0 1 0 1</td>
</tr>
<tr>
<td>-1 -1</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0 1</td>
</tr>
</tbody>
</table>

9.32 (c)

9.33 (a)

See solution for 7.10

<table>
<thead>
<tr>
<th>a b c d</th>
<th>f1 f2 f3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 1 1</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>-1 0 0 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>-1 0 0 0</td>
<td>1 0 0 1</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>0 1 0 0</td>
</tr>
</tbody>
</table>

9.33 (b)

See solution for 7.41

<table>
<thead>
<tr>
<th>x y z</th>
<th>f1 f2 f3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 0</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 1 0</td>
</tr>
</tbody>
</table>

9.33 (c)

Because a PLA works with a sum-of-products expression, see solution for 7.43(b), not (a).

9.33 (d)

See solution for 7.41

<table>
<thead>
<tr>
<th>a b c d</th>
<th>f1 f2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 1 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>-1 0 1 1</td>
<td>0 1 1</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>1 0 0</td>
</tr>
<tr>
<td>-1 0 0 1</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
9.34 \[ Z = I_A'B'C' + I_A'B'C + I_A'B'C' + I_A'BC + I_AB'C' + I_A'B'C + I_A'B'C + I_ABC \]
\[ = X_1A' + X_2A \] where \( X_1 = I_B'C' + I_B'C + I_BC' + I_BC \) and \( X_2 = I_B'C' + I_B'C + I_BC' + I_BC \)

Note: Unused inputs, outputs, and wires have been omitted from this diagram.

For an 8-to-3 encoder, using the truth table given in FLD Figure 9-16, we get

\[ a = y_4 + y_5 + y_6 + y_7 \]
\[ b = y_2y_3'y_4y_5'y_6y_7' + y_2y_3'y_4'y_5'y_6'y_7' + y_6'y_7' + y_7 \]
\[ c = y_1y_2'y_3'y_4'y_6y_7' + y_1y_3'y_4'y_6'y_7' + y_1y_3'y_6'y_7' + y_2y_4'y_5'y_7' + y_7 \]
\[ d = a + b + c + y_9 \]

Alternative solution for simplified expressions:

\[ b = y_2y_4'y_5'y_6'y_7' + y_2y_4'y_5'y_6 + y_7 \]
\[ c = y_1y_2'y_4'y_6 + y_3y_4'y_6'y_7' + y_3y_4'y_6 + y_7 \]
9.36 \( F = CD'E + CDE + A'D'E + A'B'DE' + BCD \)

9.36 (a) \( F = A'B'(CD'E + CDE + D'E + DE') + A'B (CD'E + CDE + D'E + CD) + AB'(CD'E + CDE) + AB (CD'E + CDE + CD) \)

9.36 (b) \( F = B'C' (A'D'E + A'DE') + B'C (D'E + DE + A'D'E + A'DE') + BC' (A'D'E) + BC (D'E + DE + A'D'E + D) \)

9.36 (c) \( F = A'C' (D'E + B'DE') + A'C (D'E + DE + DE + B'DE' + BD) + AC'(0) + AC(D'E + DE + BD) \)

9.36 (d) Use the expansion about \( A \) and \( C \)
\( F = A'C(F_0^*) + A'C(F_1^*) + AC(F_3^*) \)
where \( F_0^*, F_1^*, F_3^* \) are implemented in lookup tables:

\[
\begin{array}{cccc}
B & D & E & F_0^* F_1^* F_3^* \\
0 & 0 & 0 & 0 \ 0 & 1 & 1 \ 1 & 0 & 1 \ 1 & 1 & 0 \ 1 & 1 & 0 \ 1 & 1 & 1 \ 0 & 1 & 1 \ 0 & 1 & 1
\end{array}
\]
9.37  \[ F = B'D'E' + AB'C + C'DE' + A'BCD \]

9.37 (a)  \[ F = A'B' (D'E' + CDE') + A'B (C'DE' + C'D) + AB' (D'E' + C + C'D) + AB (C'DE') \]

9.37 (b)  \[ F = B'C' (D'E' + DE') + B'C (D'E' + A) + BC' (DE' + A'D) + BC (0) \]

9.37 (c)  \[ F = A'C' (B'D'E' + DE' + BD) + A'C (B'D'E' + AC (B'D'E' + DE') + AC (B'D'E' + B') \]

In this case, use the expansion about B and C to implement the function in 3 LUTs:
\[ F = B'C' (F_0) + B'C (F_1) + BC' (F_2) + BC (0) \]

Here we use the LUTs to implement \( F_0, F_1, F_2 \) which are functions of A, D, E.

9.38  For a 4-to-1 MUX:
\[ Y = A'B'I_0 + AB'I_2 + ABI_3 + AB'I_1 = A'(B'I_0 + BI_2) + A(B'I_1 + BI_3) = A'G + AF, \text{ where } G = B'I_0 + BI_2; \ F = B'I_1 + BI_3 \]

Set programmable MUX so that Y is the output of MUX H.

9.39 (a)  \[ 0 \]

9.39 (b)  \[ 0 \]

9.39 (c)  \[ 0 \]

9.40  Same answer as 9.39 except connect E to the enable input in parts (a) and (c) and E’ in part (b).
Unit 9 Solutions

9.41 (a) \[ F = a' + ac'd' + b'cd' + ad \]
\[ = d'(a' + ac'd' + b'd') + d(a' + a) \]
\[ = d'(a' + ac'd' + b'd') + d(l) \]
\[ = d'(e) + d(g) \]

9.41 (b) [Diagram of LUT 0 and LUT 1]

9.42 (a) \[ F = cd' + ad' + a'b'cd' + bc' \]
\[ = d'(c + a + bc') + d(a'b'c + bc') \]
\[ = d'(e) + d(g) \]

9.42 (b) Same as 9.41 (b).

9.43 (a) \[ F = bd + bc' + ac'd + a'd' \]
\[ = d'(be' + a') + d(b + bc' + ac') \]
\[ = d'(be' + a') + d(b + ac') \]
\[ = d'(e) + d(g) \]

9.43 (b) Same as 9.41 (b).
10.1 See FLD p. 709 for solution.  

10.2 See FLD p. 707 for solution.  

10.3 See FLD p. 710 for solution.  

10.4 See FLD p. 710 for solution.  

10.5 See FLD p. 710 for solution.  

Notes: The function vec2int is found in bit_pack, which is in the library bitlib, so the following declarations are needed to use vec2int:

```plaintext
library bitlib;
use bitlib.bit_pack.all;
```

If std_logic is used instead of bits, then the index can be computed as:

```plaintext
index <= conv_integer(A&B&Cin);
```

where A, B, and Cin are std_logic.

conv_integer is found in the std_logic_arith package.

10.6 See FLD p. 710 for solution.  

10.7 See FLD p. 711 for solution.  

Notes: In line 8, "00"&a converts a to a 18-bit std_logic_vector. The overloaded “+” operators automatically extend b, c, and d to 18 bits so that the sum is 18 bits. In line 9, sum(17 downto 2) drops the lower 2 bits of sum, which effectively divides by 4 to give the average. Adding sum(1) rounds up the value of f if sum(1) = 1.

10.8 See FLD p. 711 for solution.  

Add the following to the answer given on FLD p. 711:

```plaintext
Addout <= '0' & E + Bus;
Sum <= Addout(3 downto 0);
Cout <= Addout(4);
```

10.9 See FLD p. 711 for solution.

10.10 The network represented by the given code is:

```
  P  
   |   L
   Q ————>
   |   N
   M ————>  R
```

(1) Statement (a) will execute as soon as either P or Q change. Hence, it will execute at 4 ns.

(2) Since the NAND gate has a delay of 10 ns, L will be updated at 14 ns.

(3) Statement (c) will execute when the value M changes. It will execute at 19 ns.

(4) R will be updated at 19 + Δ ns, since Δ is the default delay time when no delay is explicitly specified.

10.11 (a) H <= not A nand B nor not D nand E;

(Note: not happens first, then it proceeds from left to right)  

10.11(b) AN <= not A after 5 ns;

C <= AN nand B after 10ns;

F <= not D after 5ns;

G <= C nor F after 15ns;

H <= G nand E after 10ns;
Unit 10 Solutions

10.12

10.13

10.14 (a) The expression can be rewritten as:
\[ F \leq (\neg E \lor "011") \land (\neg D) \]
Evaluating in this order, we get:
\[ F = 000110 \]

10.14 (b) LHS: \( \neg("101" \lor "011") = "010100" \)
RHS: \( ("100" \lor "101" \land "010" \lor "011") = "000101" \)
Since LHS > RHS, the expression evaluates to FALSE

10.15

10.16 (a) databus <= membus when mRead = '1'
else "ZZZZZZZ"

databus <= probus when mWrite = '1'
else "ZZZZZZZ"

10.16 (b) The value will be determined by the std_logic resolution function. For example, if membus = "01010101" and probus = "00001111", then databus = "0X0XX1X1"

10.17 (a) with C&D select
\[ F \leq \neg A \text{ after } 15ns \text{ when } "00", \]
\[ B \text{ after } 15ns \text{ when } "01", \]
\[ \neg B \text{ after } 15ns \text{ when } "10", \]
\[ '0' \text{ after } 15ns \text{ when } "11"; \]

10.17 (b) \[ F \leq \neg A \text{ after } 15ns \text{ when } C&D = "00" \]
else \[ B \text{ after } 15ns \text{ when } C&D = "01" \]
else \[ \neg B \text{ after } 15ns \text{ when } C&D = "10" \]
else \[ '0' \text{ after } 15ns; \]

10.18 (a) entity mynand is
\[ \text{port}(X, Y: \text{in} \text{ bit}; Z: \text{out} \text{ bit}); \]
end mynand;

architecture eqn of mynand is
begin
\[ Z \leq X \text{ nand } Y \text{ after } 4 \text{ ns}; \]
end eqn;

10.18 (b) entity main is
\[ \text{port}(A, B, C, D: \text{in} \text{ bit}; F: \text{out} \text{ bit}); \]
end main;

architecture eqn of main is
component mynand is
\[ \text{port}(X, Y: \text{in} \text{ bit}; Z: \text{out} \text{ bit}); \]
end component;

signal E, G: bit;
begin
\[ n1: \text{mynand port map}(A, B, E); \]
\[ n2: \text{mynand port map}(C, D, G); \]
\[ n3: \text{mynand port map}(E, G, F); \]
end eqn;
10.19(a) library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity hazard_circuit is
port (a, b, c : in std_logic;
f : out std_logic);
end hazard_circuit;
architecture hazarddf of hazard_circuit is
signal d, e, g : std_logic;
beg
begin
d <= not b after 10ns;
e <= a and b after 10ns;
g <= c and d after 10ns;
f <= e or g after 10ns;
end hazarddf;

10.19(b) Signal | Current | 0ns | 10ns | 20ns | 30ns | 40ns | 50ns | 60ns | 70ns | 80ns | 90ns
--------------|--------|-----|------|------|------|------|------|------|------|------|
a | '1' | 
|-----|-----|------|------|------|------|------|------|------|------|
b | '0' | 
|-----|-----|------|------|------|------|------|------|------|------|
c | '1' | 
|-----|-----|------|------|------|------|------|------|------|------|
f | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|
d | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|
e | '0' | U
|-----|-----|------|------|------|------|------|------|------|------|
g | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|

10.19(d) Signal | Current | 0ns | 10ns | 20ns | 30ns | 40ns | 50ns | 60ns | 70ns | 80ns | 90ns
--------------|--------|-----|------|------|------|------|------|------|------|------|
a | '1' | 
|-----|-----|------|------|------|------|------|------|------|------|
b | '0' | 
|-----|-----|------|------|------|------|------|------|------|------|
c | '1' | 
|-----|-----|------|------|------|------|------|------|------|------|
f | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|
d | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|
e | '0' | U
|-----|-----|------|------|------|------|------|------|------|------|
g | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|

10.19 (f) Signal | Current | 0ns | 10ns | 20ns | 30ns | 40ns | 50ns | 60ns | 70ns | 80ns | 90ns
--------------|--------|-----|------|------|------|------|------|------|------|------|
a | '1' | 
|-----|-----|------|------|------|------|------|------|------|------|
b | '0' | 
|-----|-----|------|------|------|------|------|------|------|------|
c | '1' | 
|-----|-----|------|------|------|------|------|------|------|------|
f | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|
d | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|
e | '0' | U
|-----|-----|------|------|------|------|------|------|------|------|
g | '1' | U
|-----|-----|------|------|------|------|------|------|------|------|

10.19(c) change the assignment statement for d to d <= not b after 5 ns;

10.19(e) change the assignment statement for f to f <= transport e or g after 10 ns;

10.19(g) When the output gate has a 10ns inertial delay, the 5ns glitch caused by the static-1 hazard is not passed through the gate; however, with a transport delay the glitch does pass through. Note: The initial values of d, e, f and g are 'U' because std_logic type is used. These initial values are '0' when bit type is used.
Unit 10 Solutions

10.20(a) library IEEE;
           use IEEE.STD_LOGIC_1164.ALL;
entity dynhaz_circuit is
  port (a, b, c, d : in std_logic;
        f : out std_logic);
end dynhaz_circuit;
architecture hazarddf of dynhaz_circuit is
  signal e, g, h, i, j : std_logic;
begin
  e <= not b after 10ns;
  g <= a and b after 10ns;
  h <= c and e after 10ns;
  i <= b or d after 40ns;
  j <= g or h after 10ns;
  f <= i and j after 10ns;
end hazarddf;

10.20(c) change the assignment statement for e to
e <= not b after 5 ns;

10.20(e) change the assignment statements for j and f to
j <= transport g or h after 10 ns;
f <= transport i or j after 10 ns;

10.20(g) When the gate 4 has a 10ns inertial delay, the 5ns
        glitch caused by the static-1 hazard for gate 4 is not
        passed through gate 4; however, with a transport
        delays for gates 4 and 5, the static-1 hazard glitch
        at gate 4 does pass through gate 4 and gate 5.
        In addition, the 5ns glitch caused by the delay
        through gate 3 also passes through gate 5.  The
        three changes in f illustrate the dynamic hazard that
        exists in the circuit.

### Table 10.20(b)

<table>
<thead>
<tr>
<th>Signal</th>
<th>Current</th>
<th>0ns</th>
<th>15ns</th>
<th>30ns</th>
<th>45ns</th>
<th>60ns</th>
<th>75ns</th>
<th>90ns</th>
<th>105ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 10.20(d)

<table>
<thead>
<tr>
<th>Signal</th>
<th>Current</th>
<th>0ns</th>
<th>15ns</th>
<th>30ns</th>
<th>45ns</th>
<th>60ns</th>
<th>75ns</th>
<th>90ns</th>
<th>105ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10.20(f)

<table>
<thead>
<tr>
<th>Time</th>
<th>x</th>
<th>y</th>
<th>0ns</th>
<th>15ns</th>
<th>30ns</th>
<th>45ns</th>
<th>60ns</th>
<th>75ns</th>
<th>90ns</th>
<th>105ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ns</td>
<td>0100</td>
<td>0100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 ns</td>
<td>0101</td>
<td>1011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 ns</td>
<td>1001</td>
<td>1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 ns</td>
<td>1010</td>
<td>XXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10.21(a)  
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity bcd_to_2421 is
port (x : in std_logic_vector(3 downto 0);
y : out std_logic_vector(3 downto 0));
end bcd_to_2421;

architecture behavioral1 of bcd_to_2421 is
begin
y <= "0000" when x = "0000" else "0001" when x = "0001" else "0010" when x = "0010" else "0011" when x = "0011" else "0100" when x = "0100" else "1011" when x = "0110" else "1100" when x = "0111" else "1110" when x = "1000" else "1111" when x = "1001" else "XXXX";
end behavioral1;

10.21(c)  
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity bcd_to_2421 is
port (x : in std_logic_vector(3 downto 0);
y : out std_logic_vector(3 downto 0));
end bcd_to_2421;

architecture behavioral2 of bcd_to_2421 is
begin
with x select
y <= "0000" when "0000",  "0001" when "0001",  "0010" when "0010",  "0011" when "0011",  "0100" when "0100",  "1011" when "0110",  "1100" when "0111",  "1110" when "1000",  "1111" when "1001",  "XXXX" when others;
end behavioral2;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity c8421_to_excess3 is
  port (
    x : in std_logic_vector(3 downto 0);
    y : out std_logic_vector(3 downto 0));
end c8421_to_excess3;

architecture behavioral1 of c8421_to_excess3 is
begin
  y <= "0011" when x = "0000" else "0100" when x = "0111" else "0110" when x = "0101" else "0101" when x = "0110" else "1000" when x = "0100" else "1010" when x = "0111" else "1001" when x = "1010" else "1010" when x = "1000" else "1100" when x = "1111" else "XXXX";
end behavioral1;

architecture behavioral2 of c8421_to_excess3 is
begin
  with x select
    y <= "0011" when "0000", "0100" when "0111", "0101" when "0101", "0100" when "0100", "1000" when "1011", "1001" when "1010", "1010" when "1001", "1011" when "1000", "1100" when "1111", "XXXX" when others;
end behavioral2;

<table>
<thead>
<tr>
<th>Time</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ns</td>
<td>0011</td>
<td>XXXX</td>
</tr>
<tr>
<td>5 ns</td>
<td>0100</td>
<td>0111</td>
</tr>
<tr>
<td>10 ns</td>
<td>1001</td>
<td>1010</td>
</tr>
<tr>
<td>15 ns</td>
<td>1010</td>
<td>1001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ns</td>
<td>0100</td>
<td>0111</td>
</tr>
<tr>
<td>5 ns</td>
<td>0101</td>
<td>0110</td>
</tr>
<tr>
<td>10 ns</td>
<td>1001</td>
<td>1010</td>
</tr>
<tr>
<td>15 ns</td>
<td>1010</td>
<td>1001</td>
</tr>
</tbody>
</table>
**Unit 11 Problem Solutions**

11.1 Z responds to X and to Y after 10 ns; Y responds to Z after 5 ns. See FLD p. 713 for answer.

11.2 See FLD p. 713 for solution. For part (b), also use the following Karnaugh map. Don’t cares come from the restriction in part (a).

11.3 P and Q will oscillate. See FLD p. 713 for timing chart.

11.4 See FLD p. 714 for solution.

11.5 See FLD p. 714 for solution.

11.6 (a)

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>Q⁺</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

11.6 (b) See FLD p. 714 for solution.

11.7 See FLD p. 714 for solution.

11.8 See FLD p. 714 for solution.

11.9 See FLD p. 715 for solution.

11.10 See FLD p. 715 for solution.

11.11 For every input/state combination with the condition SR = 0 holding, each circuit obeys the next-state equation \( Q⁺ = S + R'Q \). When \( S = R = 1 \), in (a), both outputs are 1, and in (b), the latch holds its state.

11.12 \( Q⁺ = R'H + HQ \)

11.13 (a)

<table>
<thead>
<tr>
<th>Present State Q</th>
<th>Next State Q⁺</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>AB</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

\( Q⁺ = AB + QA + QB \)

11.13 (b) A change between \( AB = 01 \) and 10 can cause Q to change depending on the inverter delays.

11.13 (d) \( P = Q' + A'B' \) equals \( Q' \) in all stable states.
Unit 11 Solutions

11.13 (e) A change between $AB = 01$ and 10 can cause $Q$ to change depending on the inverter delays.

$P = Q'(A' + B')$ equals $Q'$ in all stable states.

\[ Q^+ = (A + B)(A + Q) \]

11.14 (a) & (b) Present

<table>
<thead>
<tr>
<th>State</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

11.14 (c) $Q^+ = A(B' + Q)$

This is a reset dominant latch where $A'$ acts a reset and $B'$ acts as a set.

11.15 (a) $Q^+ = (M + N + G)[Q + (M + N + G)N'G']$

$= (M + N + G)[Q + N'G']$

$= (M + N + G)Q + MN'G'$

11.15 (b) & (c)

<table>
<thead>
<tr>
<th>Q</th>
<th>000</th>
<th>001</th>
<th>011</th>
<th>010</th>
<th>100</th>
<th>101</th>
<th>111</th>
<th>110</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMN</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The stable states are in bold.

11.15 (c) When $G = 1$, the circuit is always stable. When $G = 0$, $M$ and $N$ determine the state; $N = 1$ makes the state stable and with $N = 0$ the state becomes the value of $M$. There would be a restriction on $M$ and $N$ if they could cause both inputs to the output latch to be 1 when $G = 0$. This is not possible so there is no restriction.

11.15 (d) $P = Q'[N + G + M'N'G'] = Q'[N + G + M']$

For every stable state, $P = Q'$ so $P$ is usable as the complement of $Q$.

11.16 (a) $Q^+ = AB + QB$

11.16 (b) Present

<table>
<thead>
<tr>
<th>State</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The stable states are in bold.

11.16 (c) $AB = 01$ is a hold input combination, $AB = 00$ and 10 are reset input combinations, and $AB = 11$ is a set input combination. This is reset dominant latch where $S = A$ and $R = B'$. $P = Q' + B'$. In each stable state $P = Q'$ even for the input combination $AB = 10$ (SR = 11) so $P$ is usable as $Q'$. Allowing the input combination $AB = 10$ (SR = 11) would result in unreliable operation if both $A$ and $B$ could change at the same time, i.e., change to $AB = 01$ (SR = 00), because the latch could end up in either state 0 or 1 depending upon the delays in the circuit.
11.17  
(a) \( Q^+ = R(S + Q) \) if \( SR = 0 \)  
(b) \( Q^+ = (G + Q)(G' + D) \)  
(c) \( Q^+ = D \)  
(d) \( Q^+ = (Q + CE)(CE' + D) \)  
(e) \( Q^+ = (J + Q)(K' + Q') \)  
(f) \( Q^+ = (T + Q)(T' + Q') \)  

11.20 (a)  
A set-dominant FF from an S-R FF—The arrangement will ensure that when \( S = R = 1 \), \( S_1 = 1 \), \( R_1 = 0 \), and \( Q^+ = 1 \).
11.25

When $D = 0$, then $S = 0$, and $R = 1$, so $Q^+ = 0$.
When $D = 1$, then $S = 1$, and $R = 0$, so $Q^+ = 1$.

11.26

$R$ will not be ready until $D$ goes through the inverter, so we must add the delay of the inverter to the setup time:
Setup time = $1.5 + 1 = 2.5$ ns

Propagation delay for the DFF:
2.5 ns (same as for the S-R flip-flop, since the propagation delay is measured with respect to the clock)
Unit 12 Problem Solutions

12.1 Consider $3 \times Y = Y + Y + Y$, that is, we need to add $Y$ to itself 3 times. First, clear the accumulator before the first rising clock edge so that the $X$-register is 000000. Let the $Ad$ pulse be 1 for 3 rising clock edges and let the $Y$ register contain the desired number $(y_5 y_4 y_3 y_2 y_1 y_0)$ which is to be added three times. The timing diagram is on FLD p. 717. Note: $ClrN$ should go to 0 and back to 1 before the first rising clock edge. $Ad$ should be 1 before the same clock edge. However, it does not matter in what order, that is, $Ad$ could go to 1 before $ClrN$ returns to 1, or even before it goes to 0.

12.2 Serial input connected to $D_0$ for left shift. $Sh = 0, L = 1$ causes a left shift. $Sh = 1, L = 1$ or 0 causes a right shift.

12.3 See FLD Appendix E for solution.

12.4 (a) Present State $D C B A$ | Next State $D^* C^* B^* A^*$ | Flip-Flop Inputs $T_A T_B T_C T_D$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>0 0 1 0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>0 0 1 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>0 1 0 0</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>0 1 0 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>0 1 1 0</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>0 1 1 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1 0 0 0</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1 0 0 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>1 0 1 0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>1 0 1 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>1 1 0 0</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 1 0 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>1 1 1 0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>1 1 1 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>0 0 0 0</td>
<td>1 1 1 1</td>
</tr>
</tbody>
</table>

As explained in Section 12.3, it can be seen that $A$ changes on every rising clock edge: $T_A = 1$

$B$ changes only when $A = 1$: $T_B = A$

$C$ changes only when both $B$ and $A = 1$: $T_C = AB$

$D$ changes only when $A, B$, and $C = 1$: $T_D = ABC$
Unit 12 Solutions

12.4 (b) The binary counter using D flip-flops is obtained by converting each T flip-flop to a D flip-flop by adding an XOR gate.

See FLD p. 717 and Figure 12-15 on FLD p. 364.

12.5 Equations for $C$, $B$, and $A$ are from Equations (12-2) on FLD p. 364. Beginning with (b) of Problem 12.4 solutions,

\[ D' = D \oplus CBA = D'CBA + D(CBA)' \]
\[ = D'CBA + (C' + B' + A') \]
\[ = D'CBA + DC' + DB' + DA' \]

12.6 In the following state graph, the first flip-flop (C) takes on the required sequence 0, 0, 1, 0, 1, 1, (repeat).

12.7 (a) $CBA$ $C'B'A'$

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>B</th>
<th>A</th>
<th>C'B'A'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
</tbody>
</table>

For D flip-flop: 000 goes to 011 because $D_C D_B D_A = 011$

12.7 (b) $C$ $B$ $A$

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>B</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$T_C = C'A' + B'A'$

$T_B = C'B' + C B A$

$T_A = C'B A + C B' + C A'$

For T flip-flop: 000 goes to 110 because $T_A T_B T_C = 110$
12.8 (a)  

<table>
<thead>
<tr>
<th>$C$</th>
<th>$B$</th>
<th>$A$</th>
<th>$C^*B^<em>A^</em>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

In state 000,

$J_C = A' = 1, K_C = B'A' = 1, C^* = C' = 1$

$J_B = C' = 1, K_B = CA = 0, B^* = 1$

$J_A = C = 0, K_A = CB' + C'B = 0, A^* = A = 0$

So the next state is $C^*B^*A^* = 110$

12.8 (b)  

<table>
<thead>
<tr>
<th>$S$</th>
<th>$R$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>1 0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

In state 000,

$S_C = B'A' = 0, R_C = B'A' = 1, C^* = 0$

$S_B = C' = 1, R_B = CA = 0, B^* = 1$

$S_A = CA' = 0, R_A = CB' + C'BA = 0, A^* = A = 0$

So the next state is $C^*B^*A^* = 010$
See Lab Solutions for Unit 12 in this manual.
12.11 The flip-flops change state only when \( Ld \) or \( Sh \) = 1. So \( CE = Sh + Ld \). Now only a 2-to-1 MUX is required to select the input to the D flip-flop.

![Diagram](image)

12.12 (a) When \( ShLd = 00 \), the MUX for flip-flop \( i \) selects \( Q_i \) to hold its state.
When \( ShLd = 01 \), the MUX for flip-flop \( i \) selects \( D_i \) to load.
When \( ShLd = 10 \) or \( 11 \), the MUX for flip-flop \( i \) selects \( Q_{i-1} \) to shift left.

![Diagram](image)

12.12 (b) \( Q_3^* = Ld'Sh'Q_3 + LdSh'D_3 + ShQ_3 \); \( Q_2^* = Ld'Sh'Q_2 + LdSh'D_2 + ShQ_2 \); \( Q_1^* = Ld'Sh'Q_1 + LdSh'D_1 + ShQ_1 \);
\( Q_0^* = Ld'Sh'Q_0 + LdSh'D_0 + ShSI \)

12.13 Notice that \( Sh \) overrides \( Ld \) when \( Sh = Ld = 1 \)

![Diagram](image)

12.14 (a) Similar to problem 12.4 (a), \( T_E = ABCD \). \( T_D \), \( T_C \), \( T_B \) and \( T_A \) remain unchanged.

![Diagram](image)

12.14 (b) Similar to problem 12.4 (b), \( D_E = E \oplus DBCA \).
\( D_D \), \( D_C \), \( D_B \) and \( D_A \) remain unchanged.

![Diagram](image)
12.15 4-bit Johnson counter using J-K flip-flops:

Starting in 0000: 0000, 1000, 1100, 1110, 1111, 0111, 0011, (repeat) 0000, ...

Starting in 0110: 0110, 1011, 0101, 0010, 1001, 0100, 1010, 1101, (repeat) 0110, ...

12.16 When $U = 1$, $D = 0$, add 001. When $U = 0$, $D = 1$, subtract 1: add 111.
When $U = 0$, $D = 0$, no change: add 000.
$U = 1$, $D = 1$, can never occur.
So add the contents of the register to $X_2X_1X_0$, where $X_2 = X_1 = D$ and $X_0 = D + U$. (Note: to save the OR gate, let $X_0 = D$ and $C_{in} = U$.)

12.17 (a) $A B C D$ $A'B'C'D'$

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>0010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>0011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>0100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>0101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>0110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>0111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$D_A = B C D + A D'$

$D_B = B'C D + B C' + B D'$

$D_C = A'C'D + C D'$

$D_D = D'$
12.17 (b) See Table 12-7 (c) on FLD p. 374.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$J_A = B C D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$J_B = C D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$J_C = A' D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$J_D = 1$

12.17 (c) See Table 12-5 (c) on FLD p. 371.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$K_A = D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$K_B = C D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$K_C = D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$K_D = 1$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$S_A = B C D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$S_B = B'C D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$S_C = A'C'D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$S_D = D'$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$R_A = C'D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$R_A = A' D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

$R_A = B'D$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Unit 12 Solutions

12.17 (d) See Table 12-4 on FLD p. 368.

12.17 (e) Use equations to find next states for unused states.

State 1101:

\[ J_A = BCD = 0, \quad K_A = D = 1, \quad A^* = 0 \]
\[ J_B = CD = 0, \quad K_B = CD = 0, \quad B^* = B = 1 \]
\[ J_C = AD = 0, \quad K_C = D = 1, \quad C^* = 0 \]
\[ J_D = 1, \quad K_D = 1, \quad D^* = D' = 0 \]

So the next state is 0100. Other next states can be found in a similar way.

12.18

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>A'B'C'D' + AD</th>
<th>BD + BC + AD'</th>
<th>CD + BC'D' + AD'</th>
<th>D'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

12.18 (a) \[ D_A = A'B'C'D' + AD; \]
\[ D_B = BD + BC + AD'; \]
\[ D_C = CD + BC'D' + AD'; \]
\[ D_D = D' \]

12.18 (b) \[ J_A = B'C'D'; \]
\[ J_B = AD'; \]
\[ J_C = BD' + AD'; \]
\[ J_D = 1, \quad K_D = 1 \]

12.18 (c) \[ S_A = A'B'C'D', \quad R_A = AD'; \]
\[ S_B = AD', \quad R_B = BC'D' \quad \text{or} \quad A'C'D'; \]
\[ S_C = BD'C' + AD', \quad R_C = CD'; \]
\[ S_D = D', \quad R_D = D \]

12.18 (d) \[ T_A = B'C'D'; \]
\[ T_B = BC'D' + AC'D'; \]
\[ T_C = CD' + BD' + AD'; \]
\[ T_D = 1 \]

12.18 (e)
### Unit 12 Solutions

**12.19**

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$A'B'C'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

12.19 (a) $D_A = B' + AC$; $D_B = AC + BC'$; $D_C = A'B + AB'$

12.19 (b) $J_A = B', K_A = BC'$; $J_B = AC$, $K_B = A'C$; $J_C = A' + B'$, $K_C = A'B' + AB$

12.19 (c) $T_A = A'B' + ABC'$; $T_B = A'BC + AB'C'$; $T_C = A'B' + A'C' + B'C' + ABC$

12.19 (d) $S_A = B'$; $R_A = BC'$; $S_B = AC$, $R_B = A'C'$; $S_C = A'B' + AB'$, $R_C = A'B' + AB$

12.19 (e) State 000 goes to 100, because $D_A D_B D_C = 100$.

---

**12.20**

<table>
<thead>
<tr>
<th>$ABCD$</th>
<th>$D_A D_B D_C D_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>0001</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>0010</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>0011</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>0100</td>
<td>1 0 1 1</td>
</tr>
<tr>
<td>0101</td>
<td>x x x x</td>
</tr>
<tr>
<td>0110</td>
<td>x x x x</td>
</tr>
<tr>
<td>0111</td>
<td>x x x x</td>
</tr>
<tr>
<td>1000</td>
<td>x x x x</td>
</tr>
<tr>
<td>1010</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>1011</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>1101</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>1110</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>1111</td>
<td>0 0 0 0</td>
</tr>
</tbody>
</table>

12.20 (a) $\begin{align*} D_A &= AB' + A D' + BC' \\
D_B &= B'CD + A D' + AC' \\
D_C &= C'D + CD' + A'B \\
D_D &= D' \end{align*}$

12.20 (b) $\begin{align*} J_A &= BCD \\
K_A &= BCD \\
J_B &= CD \\
K_B &= A' + CD \\
J_C &= D + AB \\
K_C &= D \\
J_D &= I \\
K_D &= I \end{align*}$

12.20 (c) $\begin{align*} T_A &= A'B + BCD \\
T_B &= CD + A'B \\
T_C &= D + A'B \\
T_D &= I \end{align*}$

12.20 (d) $\begin{align*} S_A &= A'B or BC' or BD' \\
S_B &= BCD \\
S_C &= B'CD \\
S_D &= C'D + A'B \\
R_A &= BCD \\
R_B &= BCD + A'C' or BCD + A'D' or BCD + A'B \\
R_C &= CD \\
R_D &= D \end{align*}$
Unit 12 Solutions

12.21(a) \( D_A = (B' + C + D')(A + B) \)
\( D_B = (B' + C' + D')(A + D)(A + C) \) or
\( = (B' + C' + D')(B + D)(A + C) \) or
\( = (B' + C' + D')(B + C)(A + D) \)
\( D_C = (B + C + D)(C' + D')(A' + C + D) \)
\( D_D = (D') \)

12.21(b) \( J_A = (B) \)
\( K_A = (B)(C)(D) \)
\( J_B = (C)(D) \)
\( K_B = (A' + D)(A' + C) \) or
\( = (A' + C)(C' + D) \) or
\( = (A' + D)(C' + D') \)
\( J_C = (B + D)(A' + D) \)
\( K_C = (D) \)
\( J_D = (I) \)
\( K_D = (I) \)

12.21(c) \( T_A = (B)(A' + D)(A' + C) \) or
\( = (B)(A' + C)(C' + D) \) or
\( = (B)(A' + D)(C + D') \)
\( T_B = (A' + D)(B + D)(C + D') \)
or \( = (B + C)(A' + C)(C' + D) \)
\( T_B = (B + D)(A' + D) \)
\( T_B = (I) \)

12.21(d) \( S_A = (B)(D') \) or
\( = (B)(C') \) or
\( = (B)(A') \)
\( R_A = (B)(C)(D) \)
\( S_B = (C)(D)(B') \)
\( R_B = (B)(A' + D)(A' + C) \) or
\( = (B)(A' + C)(C' + D) \) or
\( = (B)(A' + D)(C' + D') \)
\( S_C = (B + D)(C')(A' + D) \)
\( R_C = (C)(D) \)
\( S_D = (D') \)
\( R_D = (D) \)

12.22(a) \[
\begin{array}{cccc|cccc}
ABCD & D_A & D_B & D_C & D_D \\
0000 & x & x & x & x \\
0001 & x & x & x & x \\
0010 & x & x & x & x \\
0011 & 0 & 1 & 0 & 0 \\
0100 & 0 & 1 & 0 & 1 \\
0101 & 0 & 1 & 1 & 0 \\
0110 & 0 & 1 & 1 & 1 \\
0111 & 1 & 0 & 0 & 0 \\
1000 & 1 & 0 & 0 & 1 \\
1001 & 1 & 0 & 1 & 0 \\
1010 & 1 & 0 & 1 & 1 \\
1011 & 1 & 1 & 0 & 0 \\
1100 & 0 & 0 & 1 & 1 \\
1101 & x & x & x & x \\
1110 & x & x & x & x \\
1111 & x & x & x & x \\
\end{array}
\]
\[
D_A = BCD + AB' \\
D_B = B'CD + A'D' + A'C' \\
D_C = C'D + CD' + AB \\
D_D = D' \\
\]

12.22(b) \[
\begin{array}{cccc|cccc|cccc}
0000 & xx & xx & xx & xx & xx & xx & xx & xx \\
0001 & xx & xx & xx & xx & xx & xx & xx & xx \\
0010 & xx & xx & xx & xx & xx & xx & xx & xx \\
0011 & 0x & 1x & x1 & x1 & 0x & 1x & x1 & x1 \\
0100 & 0x & x0 & 0x & 1x & 0x & 1x & 0x & 1x \\
0101 & 0x & x0 & 0x & 1x & 0x & 1x & 0x & 1x \\
0110 & 0x & x0 & 0x & 1x & 0x & 1x & 0x & 1x \\
0111 & 1x & x1 & x1 & x1 & 1x & x1 & x1 & x1 \\
1000 & x0 & 0x & 0x & 1x & 1x & x1 & x1 & x1 \\
1001 & x0 & 0x & 0x & 1x & 1x & x1 & x1 & x1 \\
1010 & x0 & 0x & 0x & 1x & 1x & x1 & x1 & x1 \\
1011 & x0 & 0x & 0x & 1x & 1x & x1 & x1 & x1 \\
1100 & x1 & x1 & x1 & x1 & x1 & x1 & x1 & x1 \\
1101 & xx & xx & xx & xx & xx & xx & xx & xx \\
1110 & xx & xx & xx & xx & xx & xx & xx & xx \\
1111 & xx & xx & xx & xx & xx & xx & xx & xx \\
\end{array}
\]
12.22(c) \[ T_A = AB + CD \]
\[ T_B = CD + AB \]
\[ T_C = D + AB \]
\[ T_D = 1 \]

12.23(a) \[ \begin{align*}
D_A &= (A + B)(B' + D)(A + C) \\
 &= (A + B)(B' + C)(A + D) \\
 &= (A + B)(B' + D)(B' + C)
\end{align*} \]
\[ \begin{align*}
D_B &= (B' + C' + D')(A' + D)(B + C) \\
 &= (B' + C' + D')(A' + C)(B + D) \\
 &= (B' + C' + D')(A' + D)(A' + C)
\end{align*} \]
\[ \begin{align*}
D_C &= (A + C + D)(C' + D')(B + C + D) \\
D_D &= (D')
\end{align*} \]

12.22(d) \[ \begin{align*}
S_A &= BCD \\
R_A &= BC' \\
S_B &= B'CD \\
R_B &= BCD + AC' \\
S_C &= C'D + AB \\
R_C &= CD \\
S_D &= D' \\
R_D &= D
\end{align*} \]

12.23(b) \[ \begin{align*}
J_A &= (B)(C')(D) \\
K_A &= (B) \\
J_B &= (C)(D) \\
K_B &= (C' + D)(A + C) \text{ or } (A + D)(A + C) \text{ or } (C + D')(A + D) \\
J_C &= (A + D)(B + D) \\
K_C &= (D) \\
J_D &= (I) \\
K_D &= (I)
\end{align*} \]

12.23(c) \[ \begin{align*}
T_A &= (B)(C' + D)(A + C) \text{ or } (B)(A + D)(A + C) \text{ or } (B)(C + D')(A + D) \\
T_B &= (C + D')(B + D)(A + D) \text{ or } (C' + D')(A + C)(B + C) \\
T_C &= (A + D)(B + D) \\
T_D &= (I)
\end{align*} \]

12.23(d) \[ \begin{align*}
R_A &= (B)(A) \text{ or } (B)(D') \text{ or } (B)(C') \\
S_B &= (B')(C)(D) \\
R_B &= (B)(C' + D)(A + C) \text{ or } (B)(A + D)(A + C) \text{ or } (B)(C + D')(A + D) \\
S_C &= (A + D)(C')(B + D) \\
R_C &= (C)(D) \\
S_D &= (D') \\
R_D &= (D)
\end{align*} \]

12.24 (a) The counter must clear on the next clock edge when the count is 1011 so \( ClrN = (Q3Q1Q0)' \).
(b) The counter must clear when the count reaches 1100 so \( ClrN = (Q3Q2)' \).
Unit 12 Solutions

12.25

<table>
<thead>
<tr>
<th>Q3Q2Q1Q0</th>
<th>ClrN</th>
<th>Ld</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0010</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0011</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0100</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0101</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>0110</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>0111</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1000</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1001</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1010</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1100</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1101</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1110</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1111</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The transition from state 1111 to state 0000 can be effected using Clear, Parallel Load or increment. The latter gives the simplest equations. Then ClrN = 1, Ld = Q3Q2, and P3P2P1P0 = 1011.

12.26 (a) The state 0000 can only occur between states 0001 and 1000. The resulting Karnaugh map for the S_in = Q2 ⊕ Q3 case is shown below.

(b) There are two answers: S_in = Q2 ⊕ Q3 or S_in = Q0 ⊕ Q3.

(c) The state 0000 can only occur between states 0001 and 1000. The resulting Karnaugh map for the S_in = Q2 ⊕ Q3 case is shown below.

S_in = Q3Q0Q1Q2Q3

<table>
<thead>
<tr>
<th>Q2Q3</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

S_in = Q2Q3' + Q0Q2Q3 + Q1Q2Q3 + Q0Q1Q3.'

If the circuit for S_in = Q0 ⊕ Q3 is modified, then S_in = Q0Q3' + Q0Q1Q3 + Q0Q2Q3 + Q1Q2Q3.'

12.26 (c) (contd) S_in = Q0Q3' + Q0Q1Q3 + Q0Q2Q3 + Q1Q2Q3'.

12.27 (a) All stages toggle the same as for a binary counter except when the count becomes 1001, in which case stages Q0, Q1 and Q2 respond the same as for a binary counter, but Q3 must toggle (reset). Taking into account the don’t cares, the equations become

J0 = K0 = 1
J1 = K1 = Q0
J2 = K2 = Q0Q1
J3 = Q0Q1Q2
K3 = Q0Q1Q2 + Q0Q3

12.27 (b) All stages toggle the same as for a binary counter for counts 0011 through 1011. For count 1100 stages 3 and 2 must reset and stage 1 must set while stage 0 toggles as it does for a binary counter. Taking into account the don’t cares, the equations become

J0 = K0 = 1
J1 = Q0 + Q2Q3
K1 = Q0
J2 = Q0Q3
K2 = Q0Q1 + Q2Q3
J3 = Q0Q1Q2
K3 = Q0Q1Q2 + Q2Q3
K3 can be further simplified to K3 = Q2Q3.

12.27 (c) To create a design that can be cascaded, we need to add a count enable input, CE, which is ANDed with the above equations, and terminal count output, TE, such as TE = CE(Q2Q3). TE would be connected to CE of the next counter.
12.28 (a)  

<table>
<thead>
<tr>
<th>UABC</th>
<th>$S_A R_A$</th>
<th>$S_B R_B$</th>
<th>$S_C R_C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0001</td>
<td>0x, x1</td>
<td>0x, x1</td>
<td>x1</td>
</tr>
<tr>
<td>0010</td>
<td>0x, x1</td>
<td>x1</td>
<td>10</td>
</tr>
<tr>
<td>0011</td>
<td>0x, x1</td>
<td>x0</td>
<td>x1</td>
</tr>
<tr>
<td>0100</td>
<td>x1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0101</td>
<td>x0</td>
<td>0x, x1</td>
<td>x1</td>
</tr>
<tr>
<td>0110</td>
<td>x0</td>
<td>x0</td>
<td>10</td>
</tr>
<tr>
<td>0111</td>
<td>x0</td>
<td>x0</td>
<td>x1</td>
</tr>
<tr>
<td>1000</td>
<td>0x, x1</td>
<td>0x, x1</td>
<td>10</td>
</tr>
<tr>
<td>1001</td>
<td>0x, x1</td>
<td>10</td>
<td>x1</td>
</tr>
<tr>
<td>1010</td>
<td>0x, x1</td>
<td>x0</td>
<td>10</td>
</tr>
<tr>
<td>1011</td>
<td>10</td>
<td>x1</td>
<td>x1</td>
</tr>
<tr>
<td>1100</td>
<td>x0</td>
<td>0x, x1</td>
<td>10</td>
</tr>
<tr>
<td>1101</td>
<td>x0</td>
<td>10</td>
<td>x1</td>
</tr>
<tr>
<td>1110</td>
<td>x0</td>
<td>x0</td>
<td>10</td>
</tr>
<tr>
<td>1111</td>
<td>x1</td>
<td>x1</td>
<td>x1</td>
</tr>
</tbody>
</table>

Another solution for the A FF:

$S_A = B' + C$

$R_A = UABC + U'ABC' + U'AC + U'AB'$

$S_B = I$

$R_B = U'B'C + U'BC' + UBC$

$S_C = I$

$R_C = C$
### 12.28 (b)

**CE\textsubscript{A} = UBC + U'B'C'**

**DA = A'**

**DB = B'**

**DC = C'**
12.29 (a) Present State

<table>
<thead>
<tr>
<th>MN</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>00</td>
<td>01</td>
<td>xx</td>
<td>10</td>
</tr>
<tr>
<td>01</td>
<td>10</td>
<td>00</td>
<td>10</td>
<td>xx</td>
</tr>
<tr>
<td>11</td>
<td>00</td>
<td>xx</td>
<td>xx</td>
<td>xx</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>xx</td>
<td>01</td>
<td>00</td>
</tr>
</tbody>
</table>

12.29 (b) Present State

<table>
<thead>
<tr>
<th>Present State</th>
<th>MN</th>
<th>JA</th>
<th>KA</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>00</td>
<td>0x</td>
<td>0x</td>
</tr>
<tr>
<td>01</td>
<td>1x</td>
<td>0x</td>
<td>1x</td>
</tr>
<tr>
<td>11</td>
<td>x1</td>
<td>xx</td>
<td>xx</td>
</tr>
<tr>
<td>10</td>
<td>x0</td>
<td>xx</td>
<td>x1</td>
</tr>
</tbody>
</table>

12.29 (b) Present State

<table>
<thead>
<tr>
<th>Present State</th>
<th>MN</th>
<th>O0</th>
<th>O1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>00</td>
<td>11</td>
<td>00</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>10</td>
<td>00</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>00</td>
<td>10</td>
</tr>
</tbody>
</table>

12.30 Since the FFs are changing on the negative edge of the clock, the pulses must be coincident with positive portions of the clock. Assuming the clock is symmetrical, the FFs’ propagation delay must be less than half of the clock period.

(a) The ring counter requires 8 stages: Q0, Q1, ..., Q7 and Ti = (Clk)Qi for i = 0, 1, ..., 7.

(b) The Johnson counter requires 4 stages: Q0, Q1, Q2, Q3. The count sequence will be 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001. Then T0 = (Clk)Q0Q3', T1 = (Clk)Q0Q1', T2 = (Clk)Q0Q2', T3 = (Clk)Q2Q3', T4 = (Clk)Q0Q2, T5 = (Clk)Q0'Q3, T6 = (Clk)Q1'Q3, T7 = (Clk)Q2'Q3.

(c) The binary counter requires 3 stages: Q0, Q1, Q2. The count sequence will be 000, 001, 010, 011, 100, 101, 110, 111. Then T0 = (Clk)Q0Q3', T1 = (Clk)Q0Q1', T2 = (Clk)Q0Q2', T3 = (Clk)Q0Q1Q2, T4 = (Clk)Q0Q2', T5 = (Clk)Q0Q1Q2, T6 = (Clk)Q0Q1Q2, T7 = (Clk)Q0Q1Q2.

12.31 (a) Present State

<table>
<thead>
<tr>
<th>Q</th>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Q+ = UQ + VQ'

12.31 (b) Present State

<table>
<thead>
<tr>
<th>Q Q+</th>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

12.31 (c) Present State

<table>
<thead>
<tr>
<th>Q</th>
<th>A B</th>
<th>A B</th>
<th>A B</th>
<th>A B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>01</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>00</td>
<td>01</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

Q+ = UQ + VQ'

12.31 (c) Present State

<table>
<thead>
<tr>
<th>Q</th>
<th>A B</th>
<th>A B</th>
<th>A B</th>
<th>A B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>01</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>00</td>
<td>01</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

U = A'B' + Q'

V = AQ'

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Unit 12 Solutions

12.32 (a)

<table>
<thead>
<tr>
<th>Q</th>
<th>M F</th>
<th>M F</th>
<th>M F</th>
<th>M F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
</tbody>
</table>

\[ Q^+ = F^+ + Q'M' \]

12.32 (b)

<table>
<thead>
<tr>
<th>Q</th>
<th>M F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>0x</td>
</tr>
<tr>
<td>10</td>
<td>x1</td>
</tr>
<tr>
<td>11</td>
<td>00</td>
</tr>
</tbody>
</table>

\[ M = D'Q' \]
\[ F = C' + Q' \]

12.32 (c)

<table>
<thead>
<tr>
<th>Q</th>
<th>CD</th>
<th>CD</th>
<th>CD</th>
<th>CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

12.33 (a)

<table>
<thead>
<tr>
<th>Q Q^+</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>1 1 X1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0 X0</td>
</tr>
<tr>
<td>1 0</td>
<td>1 0 1X</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0 0X</td>
</tr>
</tbody>
</table>

12.33 (b)

<table>
<thead>
<tr>
<th>A B C</th>
<th>( A'B'C' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>0 1 1</td>
<td>X X X</td>
</tr>
<tr>
<td>1 1 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 0</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

12.33 (c)

\[ L_A = B, M_A = C; L_B = A'; M_B = A' + C'; L_C = A'B'; M_C = A' \] (contd)

\[ M = D'Q' \]
\[ F = C' + Q' \]
Using Karnaugh maps:

\[ J_A = A + BD + BC, \quad K_A = 0; \quad J_B = C + D, \quad K_B = C + D; \]
\[ J_C = D', \quad K_C = D'; \quad J_D = 1, \quad K_D = 1 \]

<table>
<thead>
<tr>
<th>Clock Cycle</th>
<th>Input Data</th>
<th>EnIn</th>
<th>EnAd</th>
<th>LdAc</th>
<th>LdAd</th>
<th>Accumulator Register</th>
<th>Addend Register</th>
<th>Bus</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>18</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>Input to accumulator</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>18</td>
<td>0</td>
<td>13</td>
<td>Input to addend</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>18</td>
<td>13</td>
<td>31</td>
<td>Sum to accumulator</td>
</tr>
<tr>
<td>3</td>
<td>93</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>31</td>
<td>13</td>
<td>93</td>
<td>Input to addend</td>
</tr>
<tr>
<td>4</td>
<td>47</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>31</td>
<td>93</td>
<td>124</td>
<td>Sum to accumulator</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>124</td>
<td>93</td>
<td>22</td>
<td>Input to addend</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>124</td>
<td>22</td>
<td>146</td>
<td>Sum on bus</td>
</tr>
</tbody>
</table>

Note: Register values change after the clock edge. So a value loaded from the bus appears in the register on the next clock cycle after the load signal and bus value are present.
12.36 (c) Call the values beginning in the \(A\) & \(D\) registers \(X\) and \(Y\), respectively. We want \(C = X + Y = (X'Y)'\). Invert using \(M' = 1\) NAND \(M\). To invert a value on the right side, in register \(C\) or \(D\), we will need a 1 on the left side, in register \(A\) or \(B\). This can be accomplished using \(1 = 0\) NAND (anything.)

There are several solutions using different registers. Here is an example:

<table>
<thead>
<tr>
<th>Clock Cycle</th>
<th>(G_0)</th>
<th>(G_1)</th>
<th>(E_0)</th>
<th>(E_1)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(1) NAND (A = A' = X' \rightarrow A)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(0) NAND (B = 1 \rightarrow B)</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(B) NAND (D = 1) NAND (Y = Y' \rightarrow D)</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(A) NAND (D = (X'Y)' \rightarrow D)</td>
</tr>
</tbody>
</table>

Alternate three-cycle solution:
Use \(X + Y = X + X'Y = (X'(X'Y)')'\)

<table>
<thead>
<tr>
<th>Clock Cycle</th>
<th>(G_0)</th>
<th>(G_1)</th>
<th>(E_0)</th>
<th>(E_1)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(1) NAND (A = A' = X' \rightarrow A)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>(A) NAND (D = (X'Y)' \rightarrow D)</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(A) NAND (D = (X'(X'Y)')' = X + Y \rightarrow C)</td>
</tr>
</tbody>
</table>

12.37 (a) For bit reversal using the \(D\) inputs of the shift register: \(Sh = 0, Ld = 1\)

12.37 (b) Same as Figure 12-10 (b) on FLD p. 360, except that for the "11" input of each MUX, instead of \(SI\), \(Q_3\), \(Q_2\) or \(Q_1\), use \(Q_0\), \(Q_1\), \(Q_2\), or \(Q_3\), respectively. Also, replace \(Sh\) with \(A\) and \(Ld\) with \(B\).
13.2 Notice that this is a shift register. At each falling clock edge, \( Q_3 \) takes on the value \( Q_2 \) right before the clock edge, \( Q_2 \) takes on the value \( Q_1 \) right before the clock edge, and \( Q_1 \) takes on the value \( X \) right before the clock edge. For example, if the initial state is 000 and the input sequence is \( X = 1100 \), the state sequence is = 100, 110, 011, 001, and the output sequence is \( Z = (0)0011 \). \( Z \) is always \( Q_3 \), which does not depend on the present value of \( X \). So it's a Moore machine. See FLD p. 720 for the state graph.

13.3 (a) \[
A^+ = AK'_A + A'J_A = A(B' + X) + A'(BX' + B'X) \\
B^+ = B'J_B + BK_B = ABX + B(A' + X) \\
Z = AB
\]

### Present State

<table>
<thead>
<tr>
<th>( AB )</th>
<th>( X = 0 )</th>
<th>( X = 1 )</th>
<th>( Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>01</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

13.3 (b) \[
X = 0 \quad 1 \quad 1 \quad 0 \quad 0 \\
AB = 00 \quad 00 \quad 10 \quad 11 \quad 01 \quad 11 \\
Z = (0) \quad 0 \quad 0 \quad 1 \quad 0 \quad 1
\]

13.3 (c) See FLD p. 720 for solution.

13.4 (a) \[
Q_1^+ = D_1 \quad Q_2^+ = D_2 \quad Q_3^+ = D_3 \\
Z
\]

\( Z \) depends on the input \( X \), so this is a Mealy machine. Because there are more than 2 state variables, we cannot put the state table in Karnaugh Map order (i.e. 00, 01, 11, 10), but we can still read the next state and output from the Karnaugh map. For example, when the input is \( X = 1 \) and the state is \( Q_1Q_2Q_3 = 110 \), we can read the next state and output from the \( XQ_1Q_2Q_3 = 1110 \) position in the Karnaugh maps for \( Q_1^+, Q_2^+, Q_3^+ \), and \( Z \). So in this case, the next state is \( Q_1^+Q_2^+Q_3^+ = 101 \) and the output is \( Z = 0 \). The entire table can be derived from the Karnaugh maps in this manner. Note: We can also fill in the state table directly from the equations, without using Karnaugh maps. See FLD p. 720 for the state table and state graph.

13.4 (b - d) See FLD p. 721 for solutions.
Unit 13 Solutions

13.5 (a) Mealy machine, because the output, $Z$, depends on the input $X$ as well as the present state.

13.5 (c - d) See FLD p. 721 for solutions.

13.5 (a) 13.5 (b)

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Note: Not all Karnaugh map entries are needed. See FLD p. 721 for the state table.

13.6 (a) After a rising clock edge, it takes 4 ns for the flip-flop outputs to change. Then the ROM will take 8 ns to respond to the new flip-flop outputs. The ROM outputs must be correct at the flip-flop inputs for at least the setup time of 2 ns before the next rising clock edge. So the minimum clock period is $(4 + 8 + 2) \text{ ns} = 14 \text{ ns}$.

13.6 (b) The correct output sequence is 0101. See FLD p. 722 for the timing diagram.

13.6 (c) Read the state transition table from ROM truth table. See FLD p. 722 for the state graph and table.

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State $Q_1^+Q_2^+$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_1Q_2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X = 0$</td>
<td>$X = 1$</td>
<td></td>
</tr>
<tr>
<td>$X = 0$</td>
<td>$X = 1$</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>01</td>
<td>00</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>01</td>
</tr>
<tr>
<td>11</td>
<td>01</td>
<td>11</td>
</tr>
</tbody>
</table>

Alternate solution: Using Karnaugh map order, swap states $S_2$ and $S_3$ in the graph and table.

13.7 (a) $Q_1^+ = J_1Q_1' + K_1Q_2 = XQ_1' + XQ_2Q_1$
$Q_2^+ = J_2Q_2' + K_2Q_1 = XQ_2' + XQ_1Q_2$
$Z = XQ_2' + XQ_2$

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State $Q_1^+Q_2^+$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_1Q_2$</td>
<td>$X = 0$</td>
<td>$X = 1$</td>
</tr>
<tr>
<td>$X = 0$</td>
<td>$X = 1$</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>11</td>
</tr>
<tr>
<td>01</td>
<td>00</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>00</td>
<td>01</td>
</tr>
<tr>
<td>10</td>
<td>00</td>
<td>11</td>
</tr>
</tbody>
</table>

13.7 (b) $Z = 00011$

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
13.8 (a) \[ Q_1' = J_1 Q_1' + K_1 Q_1 = XQ_2 Q_1' + X'Q_1 \]
\[ Q_2' = J_2 Q_2' + K_2 Q_2 = XQ_1 Q_2' + X'Q_2 \]
\[ Z = XQ_2' + X'Q_2 \]

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>( Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q_1, Q_2 )</td>
<td>( X = 0 ) ( X = 1 )</td>
<td>( X = 0 ) ( X = 1 )</td>
</tr>
<tr>
<td>00</td>
<td>00 10</td>
<td>1 0</td>
</tr>
<tr>
<td>01</td>
<td>01 00</td>
<td>0 1</td>
</tr>
<tr>
<td>11</td>
<td>11 00</td>
<td>0 1</td>
</tr>
<tr>
<td>10</td>
<td>10 01</td>
<td>1 0</td>
</tr>
</tbody>
</table>

13.8 (c) \[ Z = 10110 \]

13.9 (a) \[ Q_1' = D_1 = (X_1' + X_2' + Q_1)(Q_1 + Q_2)(X_1' + Q_2) \]
\[ Q_2' = D_2 = (X_1'X_2' + Q_1)(X_1'X_2 + Q_2) \]
\[ Z = Q_1 Q_2' \]

<table>
<thead>
<tr>
<th>State ( S_0 ), ( S_1 ), ( S_2 ), ( S_3 )</th>
<th>Present State</th>
<th>Next State ( X_1', X_2' )</th>
<th>( Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>00</td>
<td>00 01 11 10</td>
<td>0</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>01</td>
<td>11 11 01 11</td>
<td>0</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>11</td>
<td>11 10 10 10</td>
<td>0</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>10</td>
<td>10 10 00 00</td>
<td>1</td>
</tr>
</tbody>
</table>

13.9 (c) \[ Z = (0)000110 \]

13.10(a) \[ Q_1' = D_1 = X_1 X_2 Q_1 + Q_1 Q_2 + X_2 Q_2 \]
\[ Q_2' = D_2 = (X_1' + X_2') Q_2 + (X_1 + X_2) Q_1' \]
\[ Z = Q_1 Q_2' \]

<table>
<thead>
<tr>
<th>State ( S_0 ), ( S_1 ), ( S_2 ), ( S_3 )</th>
<th>Present State</th>
<th>Next State ( X_1', X_2' )</th>
<th>( Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>00</td>
<td>00 01 01 01</td>
<td>0</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>01</td>
<td>01 11 11 01</td>
<td>0</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>11</td>
<td>11 11 10 11</td>
<td>0</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>10</td>
<td>00 00 10 00</td>
<td>1</td>
</tr>
</tbody>
</table>
Unit 13 Solutions

13.10(b) Notice that $Z$ depends on the input $X$, so this is a Mealy machine.

\[
Q_1^+ = J_1 Q_1 + K_1 Q_1 = XQ_1 Q_2 + XQ_1
\]

\[
Q_2^+ = J_2 Q_2 + K_2 Q_2 = XQ_2 Q_2' + XQ_2
\]

\[
Z = Q_2 \oplus X = XQ_2 Q_2' + XQ_2
\]

**Alternate solution:** Swap states $S_2$ and $S_3$.

13.10(c) $Z = (0)000110$

13.11 (a) Notice that $Z$ depends on the input $X$, so this is a Mealy machine.

\[
Q_1^+ = J_1 Q_1 + K_1 Q_1 = XQ_1 Q_2 + XQ_1
\]

\[
Q_2^+ = J_2 Q_2 + K_2 Q_2 = XQ_2 Q_2' + XQ_2
\]

\[
Z = Q_2 \oplus X = XQ_2 Q_2' + XQ_2
\]

13.11 (b) Notice that $Z$ does not depend on either input, so this is a Moore machine.

\[
Q_1^+ = X_1 Q_1 + X_2_Q_2 + X_1 Q_2
\]

\[
Q_2^+ = X_1 Q_1' + X_2 Q_1' + X_1' Q_2
\]

\[
Z = Q_1 Q_2
\]

13.12(a) Notice that $Z$ does not depend on either input, so this is a Moore machine.

\[
Q_1^+ = X_1 Q_1 + X_2 Q_2 + X_1 Q_2
\]

\[
Q_2^+ = X_1 Q_1' + X_2 Q_1' + X_1' Q_2
\]

\[
Z = Q_1 Q_2
\]
### 13.12(a) (contd)

<table>
<thead>
<tr>
<th>State</th>
<th>Present State</th>
<th>Next State $X_1, X_2$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>00</td>
<td>00 01 01 01 01</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>01</td>
<td>01 01 01 11</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>11</td>
<td>11 11 11 11 11</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>10</td>
<td>00 00 10 00 10</td>
<td>1</td>
</tr>
</tbody>
</table>

### 13.12(b)

**Clock**

- $X_1$
- $X_2$
- $Q_1$
- $Q_2$
- $Z$

Correct output: $Z = (0)00110$

### 13.13

**Clock**

- $X$
- $Q_1$
- $Q_2$
- $Q_3$
- $Z$

Correct output: $Z = 1011$

### 13.14

**Clock**

- $X$
- $Q_1$
- $Q_2$
- $Q_3$
- $Z$

Correct output: $Z = 0011$

### 13.15 (a)

<table>
<thead>
<tr>
<th>$Q_2, Q_3$</th>
<th>$XQ_1$</th>
<th>$Q_2, Q_3$</th>
<th>$XQ_1$</th>
<th>$Q_2, Q_3$</th>
<th>$XQ_1$</th>
<th>$Q_2, Q_3$</th>
<th>$XQ_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

$D_1 = Q_1^+$

$D_2 = Q_2^+$

$D_3 = Q_3^+$
Unit 13 Solutions

13.15 (a) (contd)

<table>
<thead>
<tr>
<th>State</th>
<th>Present State</th>
<th>Next State $Q_1', Q_2', Q_3'$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>000</td>
<td>001 101</td>
<td>0 1</td>
</tr>
<tr>
<td>$S_1$</td>
<td>001</td>
<td>011 111</td>
<td>0 1</td>
</tr>
<tr>
<td>$S_2$</td>
<td>010</td>
<td>110 101</td>
<td>1 0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>011</td>
<td>000 000</td>
<td>0 1</td>
</tr>
<tr>
<td>$S_4$</td>
<td>100</td>
<td>110 011</td>
<td>1 0</td>
</tr>
<tr>
<td>$S_5$</td>
<td>101</td>
<td>000 011</td>
<td>0 1</td>
</tr>
<tr>
<td>$S_6$</td>
<td>110</td>
<td>110 101</td>
<td>1 0</td>
</tr>
<tr>
<td>$S_7$</td>
<td>111</td>
<td>010 011</td>
<td>1 0</td>
</tr>
</tbody>
</table>

13.15 (b) From diagram: 0, 1, (0), 1, 0, 1
From graph: 0, 1, 1, 0, 1
(they are the same, except for the false output)

13.15 (c) Change the input on the falling edge of the clock (assuming negligible circuit delays).

13.16 (a)

<table>
<thead>
<tr>
<th>$Q_2Q_3$</th>
<th>$XQ_1$</th>
<th>$Q_2Q_3$</th>
<th>$XQ_1$</th>
<th>$Q_2Q_3$</th>
<th>$XQ_1$</th>
<th>$Q_2Q_3$</th>
<th>$XQ_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$D_1 = Q_1'$ $D_2 = Q_2'$ $D_3 = Q_3'$

13.16 (b)

<table>
<thead>
<tr>
<th>State</th>
<th>Present State</th>
<th>Next State $Q_1', Q_2', Q_3'$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>000</td>
<td>100 011</td>
<td>1 0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>001</td>
<td>000 011</td>
<td>0 1</td>
</tr>
<tr>
<td>$S_2$</td>
<td>010</td>
<td>100 000</td>
<td>1 0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>011</td>
<td>000 000</td>
<td>0 1</td>
</tr>
<tr>
<td>$S_4$</td>
<td>100</td>
<td>110 011</td>
<td>1 0</td>
</tr>
<tr>
<td>$S_5$</td>
<td>101</td>
<td>000 011</td>
<td>0 1</td>
</tr>
<tr>
<td>$S_6$</td>
<td>110</td>
<td>111 011</td>
<td>1 0</td>
</tr>
<tr>
<td>$S_7$</td>
<td>111</td>
<td>001 001</td>
<td>0 1</td>
</tr>
</tbody>
</table>
13.18 Clock Cycle

<table>
<thead>
<tr>
<th>Clock Cycle</th>
<th>Present State</th>
<th>Next State</th>
<th>Information Gathered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$Q_1Q_2 = 00$, $X = 0$ ⇒ $Z = 1$, $Q'_1Q'_2 = 01$</td>
<td>$X = 0$, $X = 1$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$Q_1Q_2 = 01$, $X = 0$ ⇒ $Z = 0$, $X = 1$ ⇒ $Z = 1$, $Q'_1Q'_2 = 11$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$Q_1Q_2 = 11$, $X = 1$ ⇒ $Z = 1$, $X = 0$ ⇒ $Z = 0$, $Q'_1Q'_2 = 10$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$Q_1Q_2 = 10$, $X = 0$ ⇒ $Z = 1$, $X = 1$ ⇒ $Z = 0$, $Q'_1Q'_2 = 00$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$Q_1Q_2 = 00$, $X = 1$ ⇒ $Z = 0$, $Q'_1Q'_2 = 10$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$Q_1Q_2 = 10$, $X = 1$ ⇒ $(Z = 0)$, $X = 0$ ⇒ $(Z = 1)$, $Q'_1Q'_2 = 11$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$Q_1Q_2 = 11$, $X = 0$ ⇒ $(Z = 0)$, $X = 1$ ⇒ $(Z = 1)$, $Q'_1Q'_2 = 01$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$Q_1Q_2 = 01$, $X = 1$ ⇒ $(Z = 1)$, $X = 0$ ⇒ $(Z = 0)$, $Q'_1Q'_2 = 00$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>$Q_1Q_2 = 00$, $X = 0$ ⇒ $(Z = 1)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Information inside parentheses was already obtained in a previous clock cycle.

13.17 (a) $Q'_1 = J_1Q'_1 + K_1Q_1$

$= (XQ'_2 + XQ'_2)Q'_1 + (X + Q'_2)Q_1$

$= XQ'_1Q'_2 + XQ_1Q_2 + XQ_1 + Q_1Q'_2$

$= XQ'_2 + XQ_1Q_2 + XQ_1 + Q_1Q'_2$

$Q'_2$ = $J_2Q'_2 + K_2Q_2$

$= XQ'_1Q'_2 + (X' + Q_1)Q_2$

$Z = Q_1Q_2$

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>$Q'_1Q'_2$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>01</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

13.17 (c) $Z = (0)01101$

13.17 (b)

The circuit is a Moore circuit. State 2 is unused.

13.16 (c)

From diagram: 1 0 1 (0) 1 1

From graph: 1 0 1 1 1

(they are the same, except for the false output)

13.16 (d)

Change the input on the falling edge of the clock (assuming negligible circuit delays).

13.16 (b)

From diagram: 1 0 1 (0) 1 1

From graph: 1 0 1 1 1

(they are the same, except for the false output)
13.19

<table>
<thead>
<tr>
<th>Clock Cycle</th>
<th>Information Gathered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( Q_1 Q_2 = 00, X_1 = 0 \Rightarrow Z = 1, Q_1^1 Q_2^1 = 10 )</td>
</tr>
<tr>
<td>2</td>
<td>( Q_1 Q_2 = 10, X_1 = 0 \Rightarrow Z = 1, Q_1^1 Q_2^1 = 01 )</td>
</tr>
<tr>
<td>3</td>
<td>( Q_1 Q_2 = 01, X_1 = 1 \Rightarrow Z = 0, X_2 = 1 \Rightarrow Z = 1, Q_1^1 Q_2^1 = 10 )</td>
</tr>
<tr>
<td>4</td>
<td>( Q_1 Q_2 = 10, X_1 = 0 \Rightarrow Z = 1, Q_1^1 Q_2^1 = 11 )</td>
</tr>
<tr>
<td>5</td>
<td>( Q_1 Q_2 = 11, X_1 = 0 \Rightarrow Z = 0, Q_1^1 Q_2^1 = 11 )</td>
</tr>
<tr>
<td>6</td>
<td>( Q_1 Q_2 = 11, X_1 = 0 \Rightarrow Z = 0, X_2 = 1 \Rightarrow Z = 1, Q_1^1 Q_2^1 = 01 )</td>
</tr>
<tr>
<td>7</td>
<td>( Q_1 Q_2 = 01, X_1 = 1 \Rightarrow Z = 0, Q_1^1 Q_2^1 = 00 )</td>
</tr>
<tr>
<td>8</td>
<td>( Q_1 Q_2 = 00, X_1 = 1 \Rightarrow Z = 1, Q_1^1 Q_2^1 = 11 )</td>
</tr>
<tr>
<td>9</td>
<td>( Q_1 Q_2 = 11, X_1 = 1 \Rightarrow (Z = 1) )</td>
</tr>
</tbody>
</table>

Note: Information inside parentheses was already obtained in a previous clock cycle.

13.20

<table>
<thead>
<tr>
<th>Clock Cycle</th>
<th>Information Gathered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( Q_1 Q_2 = 00, X_1 X_2 = 01 \Rightarrow Z, Z_1 = 10, Q_1^1 Q_2^1 = 01 )</td>
</tr>
<tr>
<td>2</td>
<td>( Q_1 Q_2 = 01, X_1 X_2 = 01 \Rightarrow Z, Z_1 = 01, X_1 X_2 = 10 \Rightarrow Z, Z_2 = 10, Q_1^1 Q_2^1 = 10 )</td>
</tr>
<tr>
<td>3</td>
<td>( Q_1 Q_2 = 10, X_1 X_2 = 10 \Rightarrow Z, Z_1 = 00, X_1 X_2 = 11 \Rightarrow Z, Z_2 = 00, Q_1^1 Q_2^1 = 01 )</td>
</tr>
<tr>
<td>4</td>
<td>( Q_1 Q_2 = 01, X_1 X_2 = 11 \Rightarrow Z, Z_1 = 11, X_1 X_2 = 01 \Rightarrow (Z, Z_2 = 01), Q_1^1 Q_2^1 = 11 )</td>
</tr>
<tr>
<td>5</td>
<td>( Q_1 Q_2 = 11, X_1 X_2 = 01 \Rightarrow Z, Z_1 = 01 )</td>
</tr>
</tbody>
</table>

Note: When \( Q_1 Q_2 = 01 \), the outputs \( Z, Z_1 \) vary depending on the inputs \( X_1 X_2 \), so this is a Mealy machine.

<table>
<thead>
<tr>
<th>Present State</th>
<th>( Q_1^1 Q_2^1 )</th>
<th>( Z, Z_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q_1 Q_2 )</td>
<td>( X_1 X_2 = )</td>
<td>( X_1 X_2 = )</td>
</tr>
<tr>
<td>00</td>
<td>00 01 11 10</td>
<td>00 01 11 10</td>
</tr>
<tr>
<td>01</td>
<td>? 11 ? 10</td>
<td>? 01 11 10</td>
</tr>
<tr>
<td>10</td>
<td>? ? 01 ?</td>
<td>? ? 00 00</td>
</tr>
</tbody>
</table>

13.21

<table>
<thead>
<tr>
<th>Present State</th>
<th>( Q_1^1 Q_2^1 )</th>
<th>( Z, Z_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q_1 Q_2 )</td>
<td>( X_1 X_2 = )</td>
<td>( X_1 X_2 = )</td>
</tr>
<tr>
<td>00</td>
<td>00 01 11 10</td>
<td>00 01 11 10</td>
</tr>
<tr>
<td>01</td>
<td>? 11 ? 10</td>
<td>? 01 11 10</td>
</tr>
<tr>
<td>10</td>
<td>? ? 01 ?</td>
<td>? ? 00 00</td>
</tr>
</tbody>
</table>

? indicates next state or output values that cannot be determined from the timing chart
13.22(a) \( Q_1^+ = D_1 = X'Q_1 + XQ_1'Q_2 \)
\( Q_2^+ = D_2 = X'Q_2 + XQ_1'Q_2' \)
\( Z = X'Q_2 + XQ_1'Q_2' + XQ_1Q_2' \)

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>( Q_1^+Q_2^+ )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q_1Q_2 )</td>
<td>( X = 0 )</td>
<td>( X = 0 )</td>
</tr>
<tr>
<td>( S_0 = 00 )</td>
<td>00 01</td>
<td>0 1</td>
</tr>
<tr>
<td>( S_1 = 01 )</td>
<td>01 10</td>
<td>1 0</td>
</tr>
<tr>
<td>( S_3 = 11 )</td>
<td>11 00</td>
<td>1 0</td>
</tr>
<tr>
<td>( S_1 = 10 )</td>
<td>10 00</td>
<td>0 1</td>
</tr>
</tbody>
</table>

13.22(c) \( Z = 11101 \)

13.23 (a) \( X_1 \)

13.23 (b) \( X_2 \)

Clock

\( Q_1 \)

\( Q_2 \)

\( Z_1 \)

\( Z_2 \)

Correct output: \( Z_1Z_2 = 10, 10, 00, 01 \)

13.23 (b) \( X_1X_2 \)

<table>
<thead>
<tr>
<th>Present State</th>
<th>( X_1X_2 )</th>
<th>( Z_1Z_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q_1Q_2 )</td>
<td>( X_1X_2 )</td>
<td>( Z_1Z_2 )</td>
</tr>
<tr>
<td>( X_1X_2 = 00 )</td>
<td>00 01 10 11</td>
<td>00 01 10 10</td>
</tr>
<tr>
<td>( X_1X_2 = 01 )</td>
<td>11 11 10 10</td>
<td>00 01 10 11</td>
</tr>
<tr>
<td>( X_1X_2 = 10 )</td>
<td>11 00 11 00</td>
<td>00 00 10 10</td>
</tr>
<tr>
<td>( X_1X_2 = 11 )</td>
<td>10 01 10 01</td>
<td>00 00 00 00</td>
</tr>
</tbody>
</table>
13.24 Transition table using a straight binary state assignment:

<table>
<thead>
<tr>
<th>State</th>
<th>Present State</th>
<th>Next State $Q_1'Q_2'Q_3'$</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>000</td>
<td>001 011</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>001</td>
<td>010 011</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>010</td>
<td>001 011</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>011</td>
<td>100 000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>100</td>
<td>011 000</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Correct output: $Z = 0, 0, 1, 0, 0, 1$

All flip-flop inputs are stable for more than the setup time before each falling clock edge. So the circuit is operating properly.

13.25 (a) If $X$ is changed early enough:

$$\text{Minimum clock period} = \text{Flip-flop propagation delay} + \text{Two NAND-gate delays} + \text{Setup time}$$

$$= 4 + (3 + 3) + 2 = 12 \text{ ns}$$

$X$ can change as late as 8 ns (two NAND-gate delays plus the setup time) before the next falling edge without causing improper operation.

13.25 (b) Correct output: $Z = 1 \ 0 \ 1 \ 0 \ 1$

Deriving the State Table:

JK flip-flop equation:

$$Q'^+ = JQ' + K'Q$$

$$\therefore \ A'^+ = (X'C + XC') A' + X'A$$

$$= A'X'C + A'XC' + X'A$$

Similarly, 

$$B'^+ = X'C + XA + X'A'C$$

$$C'^+ = 0' \cdot C + (XB'A) \cdot C' = C + XB'A$$

$$Z = XB + X'C + X'B'A$$

13.26 Deriving the State Table:

$A$  $B$  $C$  $X$  $Z$

Correct output: $Z = 1 \ 0 \ 1 \ 0 \ 1$

Deriving the State Table:

JK flip-flop equation:

$$Q'^+ = JQ' + K'Q$$

$$\therefore \ A'^+ = (X'C + XC') A' + X'A$$

$$= A'X'C + A'XC' + X'A$$

Similarly, 

$$B'^+ = X'C + XA + X'A'C$$

$$C'^+ = 0' \cdot C + (XB'A) \cdot C' = C + XB'A$$

$$Z = XB + X'C + X'B'A$$
From the Karnaugh maps, we can get the state table that follows:

<table>
<thead>
<tr>
<th>State</th>
<th>Present State</th>
<th>Next State</th>
<th>$A^+B^+C^+$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>000</td>
<td>000</td>
<td>110</td>
<td>1</td>
</tr>
<tr>
<td>$S_1$</td>
<td>001</td>
<td>111</td>
<td>001</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>010</td>
<td>000</td>
<td>110</td>
<td>1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>011</td>
<td>111</td>
<td>001</td>
<td>0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>100</td>
<td>100</td>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>$S_5$</td>
<td>101</td>
<td>101</td>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>$S_6$</td>
<td>110</td>
<td>100</td>
<td>010</td>
<td>1</td>
</tr>
<tr>
<td>$S_7$</td>
<td>111</td>
<td>101</td>
<td>011</td>
<td>0</td>
</tr>
</tbody>
</table>

13.27

$R = X_2(\bar{X}_1 + B)$

$S = \bar{X}_2(\bar{X}_1 + B)$

$A^+ = A(\bar{X}_2(\bar{X}_1 + B) + X_1(\bar{X}_1 + B))$

$A^+ = AX_2 + AX_1B + X_2X_1 + X_2B$

$T = \bar{X}_1B + X_1B'\bar{A'}$

$B^+ = BT' + B'T$

$= B(X_1B + X_1B'A')B'(X'B + X'B'A')$

$= B(\bar{X}_2(\bar{X}_1 + B) + X_1B') + X_1B'A'$

$= (BX_1 + B)X_1 + BX_1A + BA'B + X_1B'A'$

$= X_1(B + 1 + A + A') + B(1 + X_1B)$
Unit 14 Problem Solutions

14.4 Typical input and output sequences:

\[
X = \begin{array}{cccccccccccccccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\end{array} ... \\
Z = \begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array} ... \text{(output remains 1)} \\
X = \begin{array}{cccccccccccccccc}
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
\end{array} ... \\
Z = \begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array} ... \text{(output remains 1)} \\
X = \begin{array}{cccccccccccccccc}
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
\end{array} ... \\
Z = \begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array} ... \text{(output remains 1)}
\]

See FLD p. 723 for state graph.

The state meanings are given in the following table:

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Reset</td>
</tr>
<tr>
<td>$S_1$</td>
<td>One 0, no 1’s</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$\geq$ Two 0’s, no 1’s</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$\geq$ Two 0’s and one 1</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$\geq$ Two 0’s and $\geq$ Two 1’s</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$\geq$ One 1, no 0’s</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$\geq$ Two 1’s, no 0’s</td>
</tr>
<tr>
<td>$S_7$</td>
<td>$\geq$ Two 1’s and one 0</td>
</tr>
<tr>
<td>$S_8$</td>
<td>One 0 and one 1</td>
</tr>
</tbody>
</table>

14.5 Typical input and output sequence:

\[
X = \begin{array}{cccccccccccccccc}
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array} ... \\
Z_1 = \begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array} ... \text{(output remains 0 after 100 received)} \\
Z_2 = \begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array} ... \text{(at this point, the sequence 01 has occurred, so} Z_1 = 0 \text{from now on)}
\]

The graph needs two distinct parts. The first checks for 010 and 100. If 100 is received, we proceed to the second part of the graph, which checks only for 100. The two parts are joined by a one-way arc, so once in the second part it is impossible to go back to the first.

See FLD p. 723 for state table and graph.

The state meanings are given in the following table:

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Reset</td>
</tr>
<tr>
<td>$S_1$</td>
<td>Last input was 0, 100 has never occurred</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Last input was 01, 100 has never occurred</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Last input was 1, 100 has never occurred</td>
</tr>
<tr>
<td>$S_4$</td>
<td>Last input was 10, 100 has never occurred</td>
</tr>
<tr>
<td>$S_5$</td>
<td>Last input was 0, 100 has occurred at least once</td>
</tr>
<tr>
<td>$S_6$</td>
<td>Last input was 1, 100 has occurred at least once</td>
</tr>
<tr>
<td>$S_7$</td>
<td>Last input was 10, 100 has occurred at least once</td>
</tr>
</tbody>
</table>
Unit 14 Solutions

14.6 This should be solved in the same way as Example 3 on FLD p. 443. Assign a state to each possible input (00, 01, 11, 10) with an output of 0, and another state to each input with an output of 1. This gives eight states.

See FLD p. 724 for the state table.

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₀</td>
<td>Last input was 00 Z = 0</td>
</tr>
<tr>
<td>S₁</td>
<td>Last input was 01 Z = 0</td>
</tr>
<tr>
<td>S₂</td>
<td>Last input was 11 Z = 1</td>
</tr>
<tr>
<td>S₃</td>
<td>Last input was 10 Z = 1</td>
</tr>
</tbody>
</table>

Each input takes you to the state defined by that input (e.g., an input of 01 takes you to either S₁ or S₃). The only thing in question is whether the output is 0 or 1. Determine the output by checking whether the last two inputs correspond to the three input sequences.

Alternate Solution: Notice that when Z = 0, “causes the output to become 0” is the same as remaining constant, and “causes the output to become 1” is the same as toggling the output. The situation is similar when Z = 1. So we can use only four states, as follows:

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₀</td>
<td>Z = 0 and last input was either 00 or 01</td>
</tr>
<tr>
<td>S₁</td>
<td>Z = 0 and last input was either 10 or 11</td>
</tr>
<tr>
<td>S₂</td>
<td>Z = 1 and last input was either 00 or 11</td>
</tr>
<tr>
<td>S₃</td>
<td>Z = 1 and last input was either 01 or 10</td>
</tr>
</tbody>
</table>

Note: The state table with 8 states reduces to this 4-state table using methods in Unit 15.

14.7 (a) Typical input and output sequence:

X = 0 0 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 ...
Z = 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 0 1 0 ...

See FLD p. 724 for state graph.

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₀</td>
<td>Number of 1’s is divisible by three</td>
</tr>
<tr>
<td>S₁</td>
<td>Number of 1’s is one more than divisible by 3</td>
</tr>
<tr>
<td>S₂</td>
<td>Number of 1’s is two more than divisible by 3</td>
</tr>
</tbody>
</table>

14.7 (b) Typical input and output sequence:

X = 0 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 1 1 ...
Z = 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 ...

See FLD p. 724 for state graph.
<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>Number of 1’s is divisible by three, no 0’s</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>Number of 1’s is one more than divisible by 3, no 0’s</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>Number of 1’s is two more than divisible by 3, no 0’s</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>Number of 1’s is divisible by three, number of 0’s is odd</td>
</tr>
<tr>
<td>( S_4 )</td>
<td>Number of 1’s is one more than divisible by 3, number of 0’s is odd</td>
</tr>
<tr>
<td>( S_5 )</td>
<td>Number of 1’s is two more than divisible by 3, number of 0’s is odd</td>
</tr>
<tr>
<td>( S_6 )</td>
<td>Number of 1’s is divisible by three, number of 0’s is even and &lt; 0</td>
</tr>
<tr>
<td>( S_7 )</td>
<td>Number of 1’s is one more than divisible by 3, number of 0’s is even and &lt; 0</td>
</tr>
<tr>
<td>( S_8 )</td>
<td>Number of 1’s is two more than divisible by 3, number of 0’s is even and &lt; 0</td>
</tr>
</tbody>
</table>

14.8 (a)  Typical input and output sequence:

\[ X_1 = 1 0 0 1 0 0 1 1 1 0 \ldots \]
\[ X_2 = 1 0 0 0 1 1 0 0 1 1 \ldots \]
\[ Z_1 = 0'0'0'0'1'0'0'1'0'1'0' \ldots \]
\[ Z_2 = 0'1'0'0'1'0'0'0'0'1' \ldots \]

*Regardless of any value of \( N \).

See FLD p. 724 for state table.

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>Reset</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>Previous input was 00</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>Previous input was 01</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>Previous input was 10</td>
</tr>
<tr>
<td>( S_4 )</td>
<td>Previous input was 11</td>
</tr>
</tbody>
</table>

14.8 (b)  Similar to part (a), but we need a separate state for each possible output and previous input.

See FLD p. 725 for state table.

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>Reset state / current output is = 00</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>Previous input was 00 / current output is = 00</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>Previous input was 00 / current output is = 01</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>Previous input was 01 / current output is = 10</td>
</tr>
<tr>
<td>( S_4 )</td>
<td>Previous input was 01 / current output is = 01</td>
</tr>
<tr>
<td>( S_5 )</td>
<td>Previous input was 10 / current output is = 10</td>
</tr>
<tr>
<td>( S_6 )</td>
<td>Previous input was 10 / current output is = 00</td>
</tr>
<tr>
<td>( S_7 )</td>
<td>Previous input was 11 / current output is = 10</td>
</tr>
<tr>
<td>( S_8 )</td>
<td>Previous input was 11 / current output is = 00</td>
</tr>
</tbody>
</table>

14.9 (a)  

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>Previous output bit was 0</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>Previous output bit was 1</td>
</tr>
</tbody>
</table>

See FLD p. 725 for state table.
Unit 14 Solutions

14.9 (b)  
<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Output bit is 0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>Output bit is 1</td>
</tr>
</tbody>
</table>

14.9 (c) A false output occurs in NRZI just before the input NRZ goes from 1 to 0.

14.9 (d) Notice that the Moore output is delayed to the next clock cycle.

14.10  
See FLD p. 725 for solution.

14.11  
See FLD p. 726 for state graph.

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Reset</td>
</tr>
<tr>
<td>$S_1$</td>
<td>Button pressed. First full clock cycle with $Z = 1$.</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Second full clock cycle with $Z = 1$.</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Third full clock cycle with $Z = 1$.</td>
</tr>
<tr>
<td>$S_4$</td>
<td>Fourth full clock cycle with $Z = 1$.</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$X$ has not yet returned to 0.</td>
</tr>
</tbody>
</table>

14.12 (a)  
<table>
<thead>
<tr>
<th>State</th>
<th>$x = 0$ Next State</th>
<th>$x = 1$ Next State</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td>D</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>B</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>A</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Sequence ending in 1 except $x1001$</td>
</tr>
<tr>
<td>B</td>
<td>Sequence ending in 10</td>
</tr>
<tr>
<td>C</td>
<td>Sequence ending in 100</td>
</tr>
<tr>
<td>D</td>
<td>Sequence ending in 1001</td>
</tr>
<tr>
<td>E</td>
<td>Sequence ending in 001</td>
</tr>
</tbody>
</table>

14.12 (b)  
<table>
<thead>
<tr>
<th>State</th>
<th>$x = 0$ Next State</th>
<th>$x = 1$ Next State</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td>A</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Sequence ending in 1</td>
</tr>
<tr>
<td>B</td>
<td>Sequence ending in 10</td>
</tr>
<tr>
<td>C</td>
<td>Sequence ending in 100</td>
</tr>
<tr>
<td>E</td>
<td>Sequence ending in 000</td>
</tr>
</tbody>
</table>

14.13 (a)  
<table>
<thead>
<tr>
<th>State</th>
<th>$x = 0$ Next State</th>
<th>$x = 1$ Next State</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial State</td>
</tr>
<tr>
<td>2</td>
<td>1st bit was 0</td>
</tr>
<tr>
<td>3</td>
<td>1st bit was 1</td>
</tr>
<tr>
<td>4</td>
<td>1st 2 bits were 0-</td>
</tr>
<tr>
<td>5</td>
<td>1st 2 bits were 10</td>
</tr>
<tr>
<td>6</td>
<td>1st 2 bits were 11</td>
</tr>
<tr>
<td>7</td>
<td>1st 3 bits were 0-- or -00</td>
</tr>
<tr>
<td>8</td>
<td>1st 3 bits were 1-1 or 11-</td>
</tr>
</tbody>
</table>

14.13 (b) The ‘Mealy’ circuit of Part (a) is such a Moore circuit. This is possible since the output does not depend upon the fourth (least significant) bit.

14.13 (c) The ‘Mealy’ circuit of Part (a) is such a Moore circuit. This is possible since the output does not depend upon the fourth (least significant) bit.
### 14.14 (a)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>Previous 3 bits were -00</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>Previous 3 bits were 001</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>Previous 3 bits were 010</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>Previous 3 bits were 011</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>Previous 3 bits were 101</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>Previous 3 bits were 110</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Previous 3 bits were 111</td>
</tr>
</tbody>
</table>

### 14.14 (c)
The 'Mealy' circuit of Part (a) is such a Moore circuit. This is possible since the output does not depend upon the fourth (least significant) bit.

### 14.15 (a)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Initial State</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1st bit was -</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were -0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were -1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1st 3 bits were -00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1st 3 bits were --1 or -1-</td>
<td></td>
</tr>
</tbody>
</table>

### 14.15 (c)
It is not possible in this case since the output does depend upon the fourth (most significant) bit.

### 14.16 (a)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Previous 3 bits were -00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>Previous 3 bits were --1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>Previous 3 bits were -10</td>
<td></td>
</tr>
</tbody>
</table>

### 14.16 (c)
It is not possible in this case since the output does depend upon the fourth (most significant) bit.

### Unit 14 Solutions

#### 14.14 (b)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits: -000, 0-00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits: 001</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits: 010</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits: 011</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits: 011</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits: 011</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits: 011</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>Previous 4 bits: 1010</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>Previous 4 bits: 1100</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>Previous 4 bits: 1101</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>Previous 4 bits: 1110</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>Previous 4 bits: 1111</td>
<td></td>
</tr>
</tbody>
</table>

Note: A more obvious solution uses 16 states; it can be reduced to the 13 states above using the method described in Section 15.1.

#### 14.15 (b)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Valid digit</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1st bit was -</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were -0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were -1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1st 3 bits were -00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1st 3 bits were --1 or -1-</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td>Invalid digit</td>
<td></td>
</tr>
</tbody>
</table>

#### 14.16 (b)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits were --00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits were -001</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>Previous 4 bits were -1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td></td>
<td>Previous 4 bits were --1 or -11 (invalid digit)</td>
<td></td>
</tr>
</tbody>
</table>

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Unit 14 Solutions

14.17 (a) | State | Next State | \( z \) | State Meaning |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0 0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>0 0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0 0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>8</td>
<td>0 0</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0 0</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>7</td>
<td>0 0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1 1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0 0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0 1</td>
</tr>
</tbody>
</table>

State | Next State | \( z \) | State Meaning |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0 0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>0 0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0 0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>8</td>
<td>0 0</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0 0</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>7</td>
<td>0 0</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>11</td>
<td>0 0</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>1</td>
<td>0 0</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>9</td>
<td>0 0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>9</td>
<td>1 0</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

14.17 (c) It is not possible because the output depends on the value of the fourth bit, e.g., see state 8 in Part (a).

14.18 (a) | State | Next State | \( z \) | State Meaning |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1 1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1 0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0 0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>8</td>
<td>0 0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0 0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>4</td>
<td>0 0</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>6</td>
<td>0 1</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8</td>
<td>1 1</td>
</tr>
</tbody>
</table>

State | Next State | \( z \) | State Meaning |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1 1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1 0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0 0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>8</td>
<td>0 0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0 0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>4</td>
<td>0 0</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>6</td>
<td>0 1</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8</td>
<td>1 1</td>
</tr>
</tbody>
</table>

14.18 (c) It is not possible because the output depends on the value of the fourth bit, e.g., see state 2 in Part (a).

14.18 (b) | State | Next State | \( z \) | State Meaning |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1 1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1 0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0 0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>8</td>
<td>0 0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0 0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>4</td>
<td>0 0</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>6</td>
<td>0 1</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8</td>
<td>1 1</td>
</tr>
</tbody>
</table>

14.18 (c) It is not possible because the output depends on the value of the fourth bit, e.g., see state 2 in Part (a).
### 14.19 (a)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>Initial State</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1st bit was 0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1st bit was 1</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were 00</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were 01 or 10</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were 11</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1st 3 bits were -00 or 0-0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1st 3 bits were 001 or 110</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1st 3 bits were -11 or 1-1</td>
</tr>
</tbody>
</table>

### 14.19 (b)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>Initial State, Valid digit</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1st bit was 0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1st bit was 1</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were 00</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were 01 or 10</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1st 2 bits were 11</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1st 3 bits were -00 or 0-0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1st 3 bits were 001 or 110</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1st 3 bits were -11 or 1-1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td></td>
<td>Initial State, Invalid digit</td>
</tr>
</tbody>
</table>

### 14.19 (c)

It is not possible because the output depends on the value of the fourth bit, e.g., see state 7 in Part (a).

### 14.20 (a)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Previous 3 bits were -00</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>Previous 3 bits were 001</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>Previous 3 bits were 010</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>Previous 3 bits were -11</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>Previous 3 bits were 101</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>Previous 3 bits were 110</td>
</tr>
</tbody>
</table>

### 14.20 (b)

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$x = 0$</th>
<th>$x = 1$</th>
<th>$z$</th>
<th>State Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td></td>
<td>Previous 4 bits:1100</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td></td>
<td>Previous 4 bits: -001</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td></td>
<td>Previous 4 bits: -010</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>8</td>
<td>0</td>
<td></td>
<td>Previous 4 bits: 0011</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td></td>
<td>Previous 4 bits: -101</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td></td>
<td>Previous 4 bits: -110</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td></td>
<td>Previous 4 bits: -000, 0-00</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td></td>
<td>Previous 4 bits: -111</td>
</tr>
</tbody>
</table>

### 14.20 (c)

It is not possible because the output depends on the value of the most significant (fourth) bit.

### 14.21

Plot 0's horizontally. Plot 1's vertically. Receiving a 0 takes us one state to the right. Receiving a 1 takes us one state down. The output is a 1 only in the “three 0’s or more, one 1 or more” state:

![State Diagram](image)
14.22

State | Meaning
--- | ---
S₀ | Reset
S₁ | Previous input was 0 / 011 has not occurred
S₂ | Previous input was 01 / 011 has not occurred
S₃ | (No sequence) / 011 has occurred
S₄ | Previous input was 0 / 011 has occurred
S₅ | Previous input was 01 / 011 has occurred
S₆ | Previous input was 1 / 011 has not occurred
S₇ | Previous input was 10 / 011 has not occurred

* When this point in the graph is reached, 011 has been received, and we are only looking for 011 to occur again.

14.23

State | Meaning
--- | ---
S₀ | Z = 0, last input was 10 (reset)
S₁ | Z = 0, last input was 00
S₂ | Z = 0, last input was 01
S₃ | Z = 0, last input was 11
S₄ | Z = 1, last input was 10
S₅ | Z = 1, last input was 00
S₆ | Z = 1, last input was 01
S₇ | Z = 1, last input was 11

Alternate solution has 8 states, similar to problem 14.6:
14.24 (a) We need four states to describe the 1’s received, as there are four possible remainders when dividing by four. An input of 1 takes us to the next state in cyclic fashion. An input of zero leaves us in the same state.

14.24 (b) Now, expand the state graph into two dimensions: one for 1’s and the other for 0’s. We need two states to describe the zeros, odd and even.

14.25 (a) We need four states, one for each of the possible past inputs. The next state is just the one that describes that input. The output $Z_i$ is formed by adding the value of the present state to the present input. $Z_i$ is found in a similar way:

14.25 (b) The Moore version is less intuitive. Again, we need a state for each past input. We do not, however, need a state for every possible output (this would give $4 \times 4 = 16$ states) since some outputs never occur. For instance, if the last input was zero, $Z_2$ can never be 1, because anything multiplied by zero is zero. In fact, only ten states are needed:

Note: The output can never be 01. If two integers between 0 and 3 multiply to a number greater than 2, their sum is also greater than 2, i.e. $(Z_2 = 1) \Rightarrow (Z_1 = 1)$
There are two identical parts: one with an output of 0 and one with an output of 1.

State | Meaning
--- | ---
S₂, S₅ | Previous inputs were 01
S₁, S₄ | Previous input was 0
S₀, S₃ | Previous input was 1 / Reset (S₀)

There are two identical parts: one with an output of 0 and one with an output of 1.

State | Meaning
--- | ---
S₀ | Reset
S₁ | Previous input was 1
S₂ | Previous inputs were 10
S₃ | Previous inputs were 101 (first 101)
S₄ | Previous inputs were 10 (start of second 101)
S₅ | Previous inputs were 00

This is another problem similar to 14.10. Plot the number of 0’s horizontally and the number of pairs vertically:
14.28 (contd)  

<table>
<thead>
<tr>
<th>Pairs</th>
<th>0’s</th>
<th>Present State</th>
<th>Next State</th>
<th>$Z_1Z_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$S_0$</td>
<td>$S_0$</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$S_1$</td>
<td>$S_0$</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$S_2$</td>
<td>$S_6$</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>$S_3$</td>
<td>$S_6$</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>$S_4$</td>
<td>$S_6$</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>$S_5$</td>
<td>$S_6$</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>$S_6$</td>
<td>$S_0$</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>$S_7$</td>
<td>$S_0$</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$S_8$</td>
<td>$S_0$</td>
<td>1 1 1 1</td>
</tr>
</tbody>
</table>

Note: There is a seven-state solution.

14.29 0’s are plotted horizontally. 1’s are plotted vertically.

14.30

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$Z_1Z_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X = 0$</td>
<td>$X = 1$</td>
<td>$X = 0$</td>
</tr>
<tr>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_0$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_1$</td>
<td>$S_2$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_1$</td>
<td>$S_0$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Reset, 0111</td>
</tr>
<tr>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>01</td>
</tr>
<tr>
<td>$S_1$</td>
<td>011</td>
</tr>
</tbody>
</table>

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
### Unit 14 Solutions

#### 14.31

<table>
<thead>
<tr>
<th>State</th>
<th>$X_1X_2$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>$S_2$</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

#### 14.32

Example: $X = 001100110101$

$Z = 001111011101$

Note: Overlapping sequences are allowed.

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>No sequence</td>
</tr>
<tr>
<td>$S_1$</td>
<td>00</td>
</tr>
<tr>
<td>$S_2$</td>
<td>001</td>
</tr>
<tr>
<td>$S_3$</td>
<td>0011</td>
</tr>
</tbody>
</table>

#### 14.33

<table>
<thead>
<tr>
<th>State</th>
<th>Next State $X = 0$</th>
<th>$X = 1$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_2$</td>
<td>$S_4$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_3$</td>
<td>$S_4$</td>
<td>0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_4$</td>
<td>$S_6$</td>
<td>1</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_5$</td>
<td>$S_6$</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>No 1's</td>
</tr>
<tr>
<td>$S_1$</td>
<td>One 1 in first group</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Two 1's in first group</td>
</tr>
<tr>
<td>$S_3$</td>
<td>First group 11 complete, had exactly two 1's</td>
</tr>
<tr>
<td>$S_4$</td>
<td>One 1 in second group</td>
</tr>
<tr>
<td>$S_5$</td>
<td>Two 1's in second group ($Z = 1$)</td>
</tr>
<tr>
<td>$S_6$</td>
<td>&quot;Disqualified&quot; state ($Z = 0$)</td>
</tr>
</tbody>
</table>
14.34 To delay by two clock periods, we need to remember the previous two inputs. So we have four states, one for each combination of two inputs:

<table>
<thead>
<tr>
<th>State</th>
<th>Next State $X = 0$</th>
<th>Next State $X = 1$</th>
<th>$Z$ $X = 0$ $X = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>0 0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0 0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>1 1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Previous two inputs were 00</td>
</tr>
<tr>
<td>$S_1$</td>
<td>Previous two inputs were 01</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Previous two inputs were 10</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Previous two inputs were 11</td>
</tr>
</tbody>
</table>

Note: Just go to the state that represents the last two inputs.

14.35 This is the same as 14.34, except that we need to remember the last three inputs. So we have eight states:

<table>
<thead>
<tr>
<th>State</th>
<th>Next State $X = 0$</th>
<th>Next State $X = 1$</th>
<th>$Z$ $X = 0$ $X = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>0 0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0 0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_4$</td>
<td>$S_5$</td>
<td>0 0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_6$</td>
<td>$S_7$</td>
<td>0 0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>1 1</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>1 1</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$S_4$</td>
<td>$S_5$</td>
<td>1 1</td>
</tr>
<tr>
<td>$S_7$</td>
<td>$S_6$</td>
<td>$S_7$</td>
<td>1 1</td>
</tr>
</tbody>
</table>

Note: The state number expressed in binary gives the last 3 inputs.

14.36 (a) Note: The state number expressed in binary gives the last 3 inputs.

14.36 (b) 16 states are required since the last four inputs must be remembered.
Unit 14 Solutions

14.37

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$SV$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_1$</td>
<td>00 10</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_2$</td>
<td>10 00</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_1$</td>
<td>00 10</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_0$</td>
<td>10 00</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_1$</td>
<td>00 10</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_0$</td>
<td>10 01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>No bits received</td>
</tr>
<tr>
<td>$S_1$</td>
<td>One bit received</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Two bits received; Carry-in = 0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Two bits received; Carry-in = 1</td>
</tr>
<tr>
<td>$S_4$</td>
<td>Three bits received; Carry-in = 0</td>
</tr>
<tr>
<td>$S_5$</td>
<td>Three bits received; Carry-in = 1</td>
</tr>
</tbody>
</table>

14.38

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$DB$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_1$</td>
<td>00 10</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_2$</td>
<td>10 00</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_1$</td>
<td>10 00</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_0$</td>
<td>00 10</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_5$</td>
<td>11 00</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_0$</td>
<td>00 10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>No bits received</td>
</tr>
<tr>
<td>$S_1$</td>
<td>One bit received</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Two bits received; Borrow-in = 1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Two bits received; Borrow-in = 0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>Three bits received; Borrow-in = 1</td>
</tr>
<tr>
<td>$S_5$</td>
<td>Three bits received; Borrow-in = 0</td>
</tr>
</tbody>
</table>
14.39 This is similar to 14-15, and should be answered in the same way. See the solution to 14-15 for more information.

Horizontally: Number of 1's modulo 3
Vertically: Number of 0's modulo 3.

<table>
<thead>
<tr>
<th>State</th>
<th>Next State $X = 0$</th>
<th>Next State $X = 1$</th>
<th>YZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_3$, $S_4$</td>
<td>$S_1$, $S_2$</td>
<td>00</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_5$, $S_2$</td>
<td>$S_4$, $S_2$</td>
<td>01</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_5$, $S_0$</td>
<td>$S_3$, $S_6$</td>
<td>10</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_6$, $S_2$</td>
<td>$S_4$, $S_2$</td>
<td>00</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_7$, $S_4$</td>
<td>$S_5$, $S_4$</td>
<td>01</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_8$, $S_6$</td>
<td>$S_7$, $S_6$</td>
<td>10</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$S_0$, $S_2$</td>
<td>$S_1$, $S_2$</td>
<td>00</td>
</tr>
<tr>
<td>$S_7$</td>
<td>$S_0$, $S_8$</td>
<td>$S_1$, $S_8$</td>
<td>01</td>
</tr>
<tr>
<td>$S_8$</td>
<td>$S_0$, $S_0$</td>
<td>$S_1$, $S_0$</td>
<td>10</td>
</tr>
</tbody>
</table>

14.40 This problem is essentially a circular counting exercise. Pairs of 1's take you further around the state graph. Pairs can overlap, so if the last input was a 1, and the present input is a 1, you move on. If the sequence is interrupted, you branch off while you wait for the next 1. Then, you go back to the cycle of counting.

<table>
<thead>
<tr>
<th>State</th>
<th>Next State $X = 0$</th>
<th>Next State $X = 1$</th>
<th>YZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$, $S_1$</td>
<td>$S_2$, $S_1$</td>
<td>00</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_0$, $S_2$</td>
<td>$S_3$, $S_4$</td>
<td>00</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_3$, $S_4$</td>
<td>$S_5$, $S_4$</td>
<td>01</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_3$, $S_6$</td>
<td>$S_5$, $S_6$</td>
<td>01</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_5$, $S_6$</td>
<td>$S_7$, $S_6$</td>
<td>10</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_5$, $S_4$</td>
<td>$S_7$, $S_4$</td>
<td>10</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$S_7$, $S_8$</td>
<td>$S_7$, $S_8$</td>
<td>11</td>
</tr>
<tr>
<td>$S_7$</td>
<td>$S_7$, $S_6$</td>
<td>$S_7$, $S_6$</td>
<td>11</td>
</tr>
</tbody>
</table>
We notice that input $ABXX$ becomes output $AABB$.

It can be seen that it is not necessary to remember both $A$ and $B$ at once. We remember $A$ for the first two clocks and $B$ for the next two. Notice that if the output were, say, $ABAB$, we could not do this.

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_1$  $S_4$</td>
<td>0  1</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_2$  $S_3$</td>
<td>0  0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_4$  $S_4$</td>
<td>0  0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_5$  $S_6$</td>
<td>1  1</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_0$  $S_0$</td>
<td>0  0</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_2$  $S_3$</td>
<td>1  1</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$S_0$  $S_0$</td>
<td>1  1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Reset</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$A = 0$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$A = 1$</td>
</tr>
<tr>
<td>$S_3$, $S_4$</td>
<td>$B = 0$</td>
</tr>
<tr>
<td>$S_5$, $S_6$</td>
<td>$B = 1$</td>
</tr>
</tbody>
</table>

This problem is simply addition. We need a state to describe every possible sum of money entered, i.e., 0¢ to 45¢ in 5¢ intervals.

Just go to the state with the correct sum. The 25¢ state dispenses the product ($R = 1$) and resets. States above this in value cascade down to $S_5$ by giving out a nickel. When they get to $S_5$, the product is dispensed.
14.43 (a) Look at Figure 14-19, FLD p. 445, to see that
Manchester 01 gives NRZ 00
Manchester 10 gives NRZ 11
Other Manchester inputs are presumed not to occur.

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$S_1$</td>
<td>-</td>
<td>$S_0$</td>
<td>0*</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_0$</td>
<td>-</td>
<td>1</td>
<td>1*</td>
</tr>
</tbody>
</table>

* Filled in to prevent False outputs.

14.43 (b) This is the same as the Mealy, except that we need
two reset states, one with an output of zero, the
other with an output of 1. Invalid inputs never occur.

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>-</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_1$</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>1</td>
</tr>
</tbody>
</table>

14.43 (c), (d)  

Note: Moore output is delayed one clock cycle of CLOCK2.
Unit 14 Solutions

14.44

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Reset</td>
</tr>
<tr>
<td>$S_1$</td>
<td>One ring, waiting for two (or answer)</td>
</tr>
<tr>
<td>$S_2$, $S_3, S_4, S_5$</td>
<td>One, two, or three rings, respectively; waiting for four (or answer)</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Activate answering machine; wait for it to answer</td>
</tr>
</tbody>
</table>

14.45

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>$Z_iZ_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 01 10 11</td>
<td>00 01 10 11</td>
<td></td>
</tr>
<tr>
<td>$S_0$</td>
<td>$S_0, S_5$</td>
<td>$S_5, S_0$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_1, S_2$</td>
<td>$S_2, S_0$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_2, S_1$</td>
<td>$S_1, S_0$</td>
</tr>
<tr>
<td>00 11 00 00</td>
<td>00 00 00 00</td>
<td></td>
</tr>
</tbody>
</table>

14.46

In state $S_0$ there is no specification for $X_1X_2'$. This can be corrected by adding an arc for $X_1X_2'$ or changing $X_1X_2$ to $X_1$ or changing $X_1'X_2'$ to $X_2'$.

In state $S_1$ there is a conflict for $X_1X_2$. This can be corrected by changing $X_1$ to $X_1X_2'$ or changing $X_2$ to $X_1'X_2$.
15.1 (a) Implication chart after one pass:

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>C, F</td>
<td>0, 0</td>
</tr>
<tr>
<td>C</td>
<td>B, A</td>
<td>0, 0</td>
</tr>
<tr>
<td>F</td>
<td>B, F</td>
<td>1, 0</td>
</tr>
</tbody>
</table>

Complete implication chart

Reduced state table:

<table>
<thead>
<tr>
<th>State</th>
<th>Next State $X = 0$</th>
<th>Output $X = 0$</th>
<th>$X = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>C, F</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>B, A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>B, F</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

15.1 (b) \( B \equiv C \) because \( F \equiv H, \) (and also because \( C \equiv D \))

\( F \equiv H \) because \( B \equiv H, \) (and also because \( F \equiv G, \) and \( B \equiv H \) because the output differs for \( X = 0. \)

So use the sequence \( X = 100. \)

Input: \( X: \) 1 0 0

Starting in \( B: \)

State: \( (B) \) F B

Starting in \( G: \)

State: \( (G) \) H H

So \( \lambda_1(B, 100) = 010 \neq 011 = \lambda_2(G, 100), \) and \( B \neq G. \)

(Alternative: \( \lambda_1(B, 110) = 001 \neq 000 = \lambda_2(G, 110). \)

Also, \( \lambda_1(B, 00101) \neq \lambda_2(G, 00101), \) but this requires an \( X \) of length 5.

See FLD p. 727 for reduced state table.
15.3 \( a = S_0, S_5 \)
\( b = S_3 \)
\( c = S_5, S_6 \)

Since \( S_2 \) and \( S_4 \) do not have corresponding states, the circuits are not equivalent.

15.3 (b) Starting from \( S_0 \), it is not possible to reach \( S_2 \) or \( S_4 \).
So then the circuits would perform the same.

15.4 (a) \[ D = X_2X_3Q + X_1X_2Q' + X_1X_2Q' + X_2X_3Q \]
\[ Z = Q \]

15.4 (b) \[ R = X_2X_3Q + X_1X_2Q \]
\[ Z = Q \]

15.5 (a) The first row may be all 0’s, because if a column has a 1 in the first row, we can invert it so that it has a 0 in the first row without changing the number of gates. No column should be all 0’s, because that is the same as the two flip-flop case. There are only 3 columns which fit these criteria: 001, 010, and 011. No column may be used twice, because again that is the same as the two flip-flop case. So we need only check one assignment (which consists of the three columns in any order) to see whether a three flip-flop solution is better than a two flip-flop solution. One such assignment is:
0 0 0
0 1 1
1 0 1

15.5 (b) Excluding 0000, there are 7 possible columns. All possible non-repeating combinations are given below. Those with repeating rows are crossed out; 29 assignments remain to try.
15.6 (a) Group \( (S_1, S_2, S_3, S_4) \) and \( (S_5, S_6, S_7, S_8) \).

One possible assignment:
\[
\begin{align*}
S_1 & = 000, \quad S_2 = 111 \\
S_3 & = 100, \quad S_4 = 011 \\
S_5 & = 101, \quad S_6 = 010 \\
S_7 & = 001, \quad S_8 = 110
\end{align*}
\]

15.6 (b) I: \( (S_1, S_2) \) \( (S_3, S_4) \) \( (S_5, S_6) \) \( (S_7, S_8) \)

II: \( (S_1, S_2) \) \( (S_3, S_4) \) \( (S_5, S_6) \) \( (S_7, S_8) \)

Adjacencies that are satisfied are checked (✓)

One possible assignment:
\[
\begin{align*}
S_1 & = 000, \quad S_2 = 111 \\
S_3 & = 100, \quad S_4 = 011 \\
S_5 & = 101, \quad S_6 = 010 \\
S_7 & = 001, \quad S_8 = 110
\end{align*}
\]
### 15.7 (a) Guidelines:
1. \((A, D, F)\) \((C, E)\) \((A, D)\) \((C, E)\) \((B, F)\)
2. \((F, D)\times2\) \((D, B)\) \((A, C)\times2\) \((B, F)\)
3. \((A, B, D, F)\) \((C, E)\)

See FLD p. 728 for one good solution.

### 15.7 (b) See FLD p. 728 for solution.

### 15.8 (a) Guidelines:
1. \((B, D)\times2\) \((C, D)\times2\) \((A, B)\)
2. \((B, D)\) \((A, C)\) \((A, C, B)\) \((A, B, C, D)\)
3. \((A, B)\times2\) \((B, D)\times2\) \((C, D)\times2\)

Best assignment: \(A = 00\), \(B = 01\), \(C = 10\), \(D = 11\)

### 15.8 (b)

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(Q_1)</th>
<th>(Q_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\(Q_1^+\) \(Q_2^+\)

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(Q_1)</th>
<th>(Q_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\(T_1 = X_1X_2' + Q_1Q_2X_1' + Q_1'Q_2X_1 + Q_2'X_1'X_2\)

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(Q_1)</th>
<th>(Q_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\(T_2 = Q_1'Q_2X_1' + Q_1Q_2X_1\)

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(Q_1)</th>
<th>(Q_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\(Z_1 = Q_2X_1\)

### 15.9 See FLD p. 728 for solution using \(Q_1\) and \(Q_2\):

Alternate solution using \(Q_0\) \(Q_1\) and \(Q_2\):

\(D_0 = X'Q_0 + XYQ_2\)
\(D_1 = XQ_0 + YQ_2 + X'Q_1\)
\(D_2 = YQ_1 + XYQ_2\)
\(P = XYQ_0 + Y'Q_2 + XQ_1\)
\(S = X'Q_0 + XYQ_2\)
15.10 (a)  

State | Next State | Output |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X = 0</td>
<td>X = 1</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>c</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
<td>d</td>
</tr>
<tr>
<td>c</td>
<td>a</td>
<td>d</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>a</td>
</tr>
</tbody>
</table>

15.10 (b) Input: 00  
Output starting in state **c**:  
01 (state $c \rightarrow b$ state $a \rightarrow$ state $a$)  
Output starting in state **d**:  
00 (state $d \rightarrow b$ state $d \rightarrow$ state $d$)

15.11 (a)  

State | Next State | Output |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X = 0</td>
<td>X = 1</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>e</td>
<td>c</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>f</td>
</tr>
<tr>
<td>c</td>
<td>e</td>
<td>c</td>
</tr>
<tr>
<td>e</td>
<td>c</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>b</td>
<td>b</td>
</tr>
</tbody>
</table>

15.11 (b) Input: 000  
Output starting in state **a**:  
001 (state $a \rightarrow b$ state $c \rightarrow b$ state $g \rightarrow$ state $e$)  
Output starting in state **b**:  
000 (state $b \rightarrow b$ state $d \rightarrow b$ state $b \rightarrow b$ state $d$)

15.12 (a) Equivalent States: $S_0 = S_8$, $S_2 = S_{10}$, $S_3 = S_{11}$,  
$S_4 = S_{12}$, $S_7 = S_{15}$.

15.12 (b) New Equivalent States: $S_1 = S_5$, $S_9 = S_{13}$.

<table>
<thead>
<tr>
<th>Input</th>
<th>Present State</th>
<th>Next State</th>
<th>Output</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>-000</td>
<td>$S_0$</td>
<td>$S_3$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>-010</td>
<td>$S_2$</td>
<td>$S_4$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>-011</td>
<td>$S_3$</td>
<td>$S_6$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-100</td>
<td>$S_4$</td>
<td>$S_0$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>$S_5$</td>
<td>$S_2$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>$S_6$</td>
<td>$S_4$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-111</td>
<td>$S_7$</td>
<td>$S_14$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>$S_9$</td>
<td>$S_2$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>$S_13$</td>
<td>$S_2$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>$S_{14}$</td>
<td>$S_4$</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
### 15.12 (c) Present Next State Output Z

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Input</th>
<th>Present State</th>
<th>Next State</th>
<th>Output Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$X = 0$</td>
<td>$X = 1$</td>
</tr>
<tr>
<td>-000</td>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>0-01</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0</td>
</tr>
<tr>
<td>-010</td>
<td>$S_2$</td>
<td>$S_4$</td>
<td>$S_1$</td>
<td>1</td>
</tr>
<tr>
<td>-011</td>
<td>$S_3$</td>
<td>$S_6$</td>
<td>$S_7$</td>
<td>1</td>
</tr>
<tr>
<td>-100</td>
<td>$S_4$</td>
<td>$S_0$</td>
<td>$S_9$</td>
<td>0</td>
</tr>
<tr>
<td>0110</td>
<td>$S_6$</td>
<td>$S_4$</td>
<td>$S_9$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$-111$</td>
<td>$S_7$</td>
<td>$S_{14}$</td>
<td>0</td>
</tr>
<tr>
<td>1-01</td>
<td>$S_9$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0</td>
</tr>
<tr>
<td>1110</td>
<td>$S_{14}$</td>
<td>$S_4$</td>
<td>$S_9$</td>
<td>1</td>
</tr>
</tbody>
</table>

### 15.12 (d) $S_1 = S_9$ and $S_6 = S_{14}$

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Input</th>
<th>Present State</th>
<th>Next State</th>
<th>Output Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$X = 0$</td>
<td>$X = 1$</td>
</tr>
<tr>
<td>-000</td>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>-01</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0</td>
</tr>
<tr>
<td>-010</td>
<td>$S_2$</td>
<td>$S_4$</td>
<td>$S_1$</td>
<td>1</td>
</tr>
<tr>
<td>-011</td>
<td>$S_3$</td>
<td>$S_6$</td>
<td>$S_7$</td>
<td>1</td>
</tr>
<tr>
<td>-100</td>
<td>$S_4$</td>
<td>$S_0$</td>
<td>$S_9$</td>
<td>0</td>
</tr>
<tr>
<td>-110</td>
<td>$S_6$</td>
<td>$S_4$</td>
<td>$S_9$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$-111$</td>
<td>$S_7$</td>
<td>$S_{14}$</td>
<td>0</td>
</tr>
<tr>
<td>1-01</td>
<td>$S_9$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0</td>
</tr>
<tr>
<td>1110</td>
<td>$S_{14}$</td>
<td>$S_4$</td>
<td>$S_9$</td>
<td>1</td>
</tr>
</tbody>
</table>

### 15.13 (a) Moore circuit.

### 15.13
- $S_8 = S_9 = S_{10} = S_{11} = S_{12}$
- $S_{13} = S_{14} = S_{15}$

### 15.14 (a) Present Next State Output Z

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Input</th>
<th>Present State</th>
<th>Next State</th>
<th>Output Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$X = 0$</td>
<td>$X = 1$</td>
</tr>
<tr>
<td>-</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>$S_2$</td>
<td>$S_4$</td>
<td>$S_5$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$S_3$</td>
<td>$S_6$</td>
<td>$S_7$</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>$S_4$</td>
<td>$S_8$</td>
<td>$S_9$</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>$S_5$</td>
<td>$S_{10}$</td>
<td>$S_{11}$</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>$S_6$</td>
<td>$S_{12}$</td>
<td>$S_{13}$</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>$S_7$</td>
<td>$S_{14}$</td>
<td>$S_{15}$</td>
<td>0</td>
</tr>
<tr>
<td>000</td>
<td>$S_8$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>$S_9$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>$S_{10}$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>$S_{11}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>$S_{12}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>$S_{13}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>110</td>
<td>$S_{14}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>$S_{15}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
</tbody>
</table>

### 15.14 (b) Present Next State Output Z

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Input</th>
<th>Present State</th>
<th>Next State</th>
<th>Output Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$X = 0$</td>
<td>$X = 1$</td>
</tr>
<tr>
<td>-</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>$S_2$</td>
<td>$S_4$</td>
<td>$S_5$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$S_3$</td>
<td>$S_6$</td>
<td>$S_7$</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>$S_4$</td>
<td>$S_8$</td>
<td>$S_9$</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>$S_5$</td>
<td>$S_{10}$</td>
<td>$S_{11}$</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>$S_6$</td>
<td>$S_{12}$</td>
<td>$S_{13}$</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>$S_7$</td>
<td>$S_{14}$</td>
<td>$S_{15}$</td>
<td>0</td>
</tr>
<tr>
<td>000</td>
<td>$S_8$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>$S_9$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>$S_{10}$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>$S_{11}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>$S_{12}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>$S_{13}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>110</td>
<td>$S_{14}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>$S_{15}$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
</tbody>
</table>
15.14(b), Equivalent States: $S_4 = S_6$ and $S_5 = S_7$.

(c) (contd)

<table>
<thead>
<tr>
<th>Input Pattern</th>
<th>Present State</th>
<th>Next State $X = 0$</th>
<th>Next State $X = 1$</th>
<th>Output $Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>$S_2$</td>
<td>$S_4$</td>
<td>$S_5$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$S_3$</td>
<td>$S_4$</td>
<td>$S_5$</td>
<td>0</td>
</tr>
<tr>
<td>-0</td>
<td>$S_4$</td>
<td>$S_8$</td>
<td>$S_9$</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>$S_5$</td>
<td>$S_8$</td>
<td>$S_9$</td>
<td>0</td>
</tr>
<tr>
<td>-00</td>
<td>$S_8$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>-01, -1-</td>
<td>$S_9$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
</tbody>
</table>

15.15 (a)

$$\begin{align*}
\text{Present State} & \quad \text{Next State} \\
00 & \quad 0 \\
01 & \quad 1 \\
11 & \quad 1 \\
10 & \quad 0
\end{align*}$$

$$
\begin{array}{cccc}
\text{Pattern State} & X = 0 & X = 1 & \text{Output Z} \\
\hline
X = 0 & S_1 & S_2 & S_3 & 0 & 0 \\
X = 1 & S_2 & S_4 & S_5 & 0 & 0 \\
X = 0 & S_4 & S_8 & S_9 & 0 & 0 \\
X = 1 & S_5 & S_8 & S_9 & 0 & 0 \\
X = 0 & S_8 & S_1 & S_1 & 0 & 0 \\
X = 1 & S_9 & S_1 & S_1 & 0 & 1
\end{array}
$$

15.15 (b)

$$
\begin{array}{cccc}
\text{State} & \text{Next State} & X = 0 & X = 1 \\
\hline
a & b & c & 0 \\
b & e & b & 1 \\
c & g & b & 1 \\
e & c & g & 1 \\
g & g & i & 0 \\
i & a & a & 0 \\
i & a & a & 1
\end{array}
$$
Unit 15 Solutions

15.16 (a)

Present State | Next State | Z
---|---|---
00 01 11 10 | 00 01 11 10 | 0
---|---|---
a | b | c | d | e | f | g | h | i | j
---|---|---|---|---|---|---|---|---|---
00 | b | f | c | g | 0
01 | b | c | f | g | 0
11 | c | a | d | f | 1
10 | d | a | c | b | g | 1
---|---|---|---|---|---|---|---|---|---
f | f | f | f | d | 0
---|---|---|---|---|---|---|---|---|---
g | a | d | g | a | 0
---|---|---|---|---|---|---|---|---|---

(b)

Present State | Next State | Z
---|---|---
00 01 11 10 | 00 01 11 10 | 0
---|---|---
---|---|---|---|---|---|---|---|---|---|---
a | a | a | a | g | b | 1 | 0 | 0 | 0
01 | b | c | e | g | a | 0 | 0 | 0 | 0
11 | c | g | e | a | b | 1 | 0 | 0 | 0
10 | g | c | a | g | b | 0 | 1 | 0 | 0
---|---|---|---|---|---|---|---|---|---|---
168

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
15.17 (a) $S_e = e = f, S_c = c = d, S_a = S_b = a = b$

Since every state in $N$ has an equivalent state in $M$, and vice versa, $N$ and $M$ are equivalent.

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
</tbody>
</table>

15.18 (a) Set don't care to $S_a$ so $S = S_b$:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S = S_b$</td>
<td>$S = S_b$</td>
<td>0</td>
</tr>
<tr>
<td>$S = S_b$</td>
<td>$S = S_b$</td>
<td>0</td>
</tr>
<tr>
<td>$S = S_b$</td>
<td>$S = S_b$</td>
<td>1</td>
</tr>
<tr>
<td>$S = S_b$</td>
<td>$S = S_b$</td>
<td>0</td>
</tr>
</tbody>
</table>

15.19 (a) Set don't care to 0 so $S_a = S_b$:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_a = 0$</td>
<td>$S_b = 1$</td>
<td>0</td>
</tr>
<tr>
<td>$S_b = 1$</td>
<td>$S_a = 0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_a = 0$</td>
<td>$S_b = 1$</td>
<td>0</td>
</tr>
<tr>
<td>$S_b = 1$</td>
<td>$S_a = 0$</td>
<td>1</td>
</tr>
</tbody>
</table>

15.17 (b)

<table>
<thead>
<tr>
<th>$X = 0$</th>
<th>$Y = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X = 0$</td>
<td>$Y = 0$</td>
</tr>
</tbody>
</table>

15.19 (b)

No equivalent states.

15.19 (c) $X = 0$

$Z = 0$

$Z' = 0$
15.20 (a) Invert all three columns of assignment (iv), and then swap the first and last columns. Then (iii) and (iv) are the same, therefore, Assignment (iii) ≡ Assignment (iv).

15.20 (c) Many state assignments are not equivalent to (i) through (v), for example:

\[
\begin{array}{ll}
101 & 011 \\
000 & 101 \\
011 & 000 \\
100 & 100 \\
010 & 010 \\
110 & 110 \\
\end{array}
\]

15.21 (a) Straight Binary Assignment

<table>
<thead>
<tr>
<th>Equivalent State Assignments (any three)</th>
<th>(c_2\leftrightarrow c_3)</th>
<th>(c_1\leftrightarrow c_3)</th>
<th>(c_1\leftrightarrow c_2)</th>
<th>(c_1\rightarrow c_3\rightarrow c_2\rightarrow c_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
<td>000</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>100</td>
<td>010</td>
<td>010</td>
</tr>
<tr>
<td>010</td>
<td>100</td>
<td>010</td>
<td>001</td>
<td>100</td>
</tr>
<tr>
<td>011</td>
<td>010</td>
<td>100</td>
<td>110</td>
<td>011</td>
</tr>
<tr>
<td>100</td>
<td>110</td>
<td>110</td>
<td>001</td>
<td>101</td>
</tr>
<tr>
<td>101</td>
<td>010</td>
<td>010</td>
<td>110</td>
<td>011</td>
</tr>
<tr>
<td>110</td>
<td>110</td>
<td>111</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>111</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>111</td>
</tr>
</tbody>
</table>

15.21 (b) Many state assignments are not equivalent to the straight binary assignment, for example:

\[
\begin{array}{ll}
111 & 111 \\
101 & 001 \\
110 & 010 \\
100 & 011 \\
011 & 100 \\
010 & 101 \\
001 & 000 \\
000 & 110 \\
\end{array}
\]

15.22 (a) 1. \((A, H) (B, G) (A, D) (E, G)\)
2. \((D, G) (E, H) (B, F) (F, G) (C, A) (H, C) (E, A) (D, B)\)
3. \((A, C, E, G) (B, D, F, H)\)

Consider Guideline #3 only:

\[
\begin{array}{ccc}
Q_1 & Q_2 & Q_3 \\
00 & B & A \\
01 & D & C \\
11 & F & E \\
10 & H & G \\
\end{array}
\]

\[
\begin{array}{ccc}
Q_1 & Q_2 & Q_3 \\
00 & 0 & 1 \\
01 & 0 & 1 \\
11 & 0 & 1 \\
10 & 0 & 1 \\
\end{array}
\]

\[Z = Q_1\]
15.22 (b) Consider Guidelines #1, 2:
A = 000, B = 111, C = 110, D = 001, E = 010,
F = 101, G = 011, H = 100

15.23 (a) 1. (A, C) × 2 ✓ (B, C) × 2 ✓ (A, D) ✓
2. (A, C) ✓ (B, D) ✓ (A, B, D) ✓
(A, B, C, D) ✓
3. (A, D) ✓
Adjacencies that are satisfied are checked (✓)

15.23 (b) (contd)
Unit 15 Solutions

15.24 (a) Equations for one-hot state assignment:

\[ D_A = X(A + B + D + E), \quad D_B = X(A + D), \]
\[ D_C = X'B, \quad D_D = XC, \quad D_E = X'(C + E), \quad z = D \]

15.24 (b) Guidelines:

1. \((A, D)\)\(\lor\) \((C, E)\) \((A, B, D, E)\)
2. \((A, B)\)\(\lor\) \((A, C)\) \((D, E)\) \((A, E)\)

The following assignment satisfies all but \((A, E)\), \((A, C)\) and \((B, D)\):

<table>
<thead>
<tr>
<th>Q₂, Q₃, Q₁</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>F</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>G</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>I</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

15.25 (a) B

15.25 (b) 1. \((A, C)\)\(\checkmark\) \((B, D)\)\(\checkmark\) \((C, E)\)\(\checkmark\)
2. \((A, B)\)\(\checkmark\) \((C, E)\)\(\checkmark\) \((A, D)\)\(\checkmark\) \((A, C)\)\(\checkmark\) \((B, D)\)\(\checkmark\)
3. \((A, C, E)\)\(\checkmark\) \((B, D)\)\(\checkmark\)

Adjacencies that are satisfied are checked (\(\checkmark\))

\(A = 000, B = 100, C = 001, D = 101, E = 011\)

All are satisfied except \((A, D)\)

15.25 (c)

<table>
<thead>
<tr>
<th>(Q₂, Q₃, Q₁)</th>
<th>(Q₂^* Q₃^* Q₁^*)</th>
<th>(X = 0)</th>
<th>(Z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>010</td>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>001</td>
<td>011</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>100</td>
<td>000</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>100</td>
<td>000</td>
<td>0</td>
</tr>
</tbody>
</table>

Alternate:

<table>
<thead>
<tr>
<th>Q₂, Q₃, Q₁</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>F</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>G</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

15.25 (c)
Output Z equation is the same for D and J-K flip-flops.
(Actually, it is the same for any flip-flop.)
Unit 15 Solutions

15.26 (b) 1. \((A, D) \times 2\)
2. \((A, C) \times 2\) \((C, G)\) \((D, E) \times 2\)
3. \((A, B, C)\) \((D, E, G)\)

There are several solutions. Here is one satisfying all guidelines:

\[A = 000,\quad B = 010,\quad C = 001,\quad D = 100,\quad E = 110,\quad G = 101\]

15.26 (c)

<table>
<thead>
<tr>
<th>(Q_1Q_2Q_3)</th>
<th>(Q_1'Q_2'Q_3')</th>
<th>(X = 0)</th>
<th>(1)</th>
<th>(Z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000 001</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>010 000</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>001 101</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>000 001</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>100 110</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>110 100</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[D_1 = Q_1Q_3 + Q_2Q_3\]

\[D_2 = X'Q_1Q_2 + XQ_1Q_3 + XQ_2Q_3\]

\[D_3 = Q_1Q_3 + XQ_2Q_3'\]

15.26 (d) Again, \(Z = Q_1'\):

\[J_1 = XQ_3\]

\[J_2 = X'Q_1Q_3\]

\[J_3 = XQ_2'\]

\[K_1 = Q_2Q_3'\]

\[K_2 = XQ_1 + XQ_1'\]

\[K_3 = Q_1\]
15.27 (a) Present State | Next State | Output
--- | --- | ---
$S_0$ | $S_1$ $S_2$ | 0 0
$S_1$ | $S_1$ $S_2$ | 1 0
$S_2$ | $S_1$ $S_2$ | 0 0
$S_3$ | $S_1$ $S_2$ | 0 0
$S_4$ | $S_1$ $S_2$ | 0 1
$S_5$ | $S_1$ $S_2$ | 0 1

Unit 15 Solutions

(a) $D_1 = XQ_1'Q_2' + Q_2 + XQ_0Q_3'$; $D_2 = XQ_1'; D_3 = XQ_2'$; $Z = XQ_1'Q_2 + XQ_0Q_3'$

(b) $S_1 = XQ_0'Q_2' + Q_3'; R_1 = XQ_1'Q_2' + Q_3'; S_2 = XQ_0'Q_2' + Q_3'; R_2 = XQ_1'; S_3 = XQ_0'Q_2'; R_3 = XQ_1'; Z = XQ_1'Q_2 + XQ_0Q_3'$

One alternative assignment:

<table>
<thead>
<tr>
<th>$Q_2$</th>
<th>$Q_3$</th>
<th>$Q_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

175

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
Guideline 3 is of no use for this state table.
15.29  
See solutions to 14.22 for the state table.
I. \((S_0, S_1, S_2) (S_3, S_4, S_5) (S_6, S_7, S_8) (S_{9}, S_{10}, S_{11}) \) 
II. \((S_1, S_0) (S_1, S_3) + 2 (S_3, S_1) (S_2, S_4) \times 2 \) 
III. \((S_0, S_1, S_2, S_3) (S_4, S_5) \) 

\[
\begin{array}{c|c|c}
Q_1 & Q_2 & Q_3 \\
\hline
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
Q_1 & Q_2 & Q_3 \\
\hline
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
0 & 0 & 0 \\
1 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
Q_1 & Q_2 & Q_3 \\
\hline
0 & 1 & 1 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c}
Q_1 & Q_2 & Q_3 & X=0 & X=1 \\
\hline
000 & 001 & 100 & 00 & 00 \\
001 & 001 & 110 & 00 & 00 \\
110 & 001 & 011 & 00 & 01 \\
011 & 111 & 011 & 00 & 00 \\
101 & 111 & 011 & 00 & 01 \\
010 & 001 & 110 & 10 & 00 \\
\end{array}
\]

15.30  
See FLD p. 723 for the state table.
I. \((S_0, S_1, S_2) (S_3, S_4) (S_5, S_6, S_7) (S_8, S_9, S_{10}) (S_{11}, S_{12}, S_{13}) \) 
II. \((S_1, S_0) (S_1, S_3) (S_3, S_1) + 2 (S_3, S_1) (S_2, S_4) \times 2 \) 
III. \((S_0, S_1, S_2, S_3) (S_4, S_5, S_6) \) 

\[
S_0 = 000, S_1 = 001, S_2 = 010, S_3 = 011, S_4 = 111, S_5 = 110, S_6 = 100, S_7 = 101
\]

\[
\begin{array}{c|c|c|c|c|c}
Q_1 & Q_2 & Q_3 & X=0 & X=1 \\
\hline
000 & 001 & 011 & 00 & 00 \\
001 & 001 & 010 & 00 & 00 \\
010 & 111 & 011 & 10 & 00 \\
011 & 111 & 011 & 00 & 00 \\
111 & 110 & 010 & 01 & 00 \\
110 & 110 & 100 & 00 & 00 \\
100 & 101 & 100 & 00 & 00 \\
101 & 110 & 100 & 01 & 00 \\
\end{array}
\]
Unit 15 Solutions

15.30 (contd)

\[ Q_1^* \]
\[ J_1 = X'Q_2 \]
\[ K_1 = XQ_3 \]
\[ Z_1 = XQ_1Q_2Q_3' \]

\[ Q_2^* \]
\[ J_2 = XQ_3 + XQ_2' \]
\[ K_2 = XQ_3' \]
\[ Z_2 = XQ_1Q_2 \]

\[ Q_3^* \]
\[ J_3 = Q_1^* + XQ_2' \]
\[ K_3 = Q_1 + XQ_2' \]

15.31 Row reduction of the solution to 14.6 given on FLD p. 724 easily gives 4 states. Renaming them gives:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>( Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>( S_0 )</td>
<td>0</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>( S_0 )</td>
<td>0</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( S_0 )</td>
<td>0</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>( S_2 )</td>
<td>1</td>
</tr>
</tbody>
</table>

See p. 146 in this manual for the state table.

- I. \((S_0, S_2) \times 3 (S_2, S_0) \times 2 (S_0, S_2) (S_1, S_2) (S_2, S_1)\)
- II. \((S_0, S_2) (S_0, S_2) (S_0, S_2) (S_2, S_1) (S_2, S_2)\)
- III. \((S_0, S_2) (S_2, S_2)\)

\( S_0 = 00, S_1 = 01, S_2 = 10, S_3 = 11 \)

\[ Q_0 \]
\[ Q_1 \]
\[ Q_2 \]
\[ Q_3 \]

\[ Z = Q_1 \]

\[ D_1 = X_1X_2Q_2 + X_1Q_1 + X_2Q_1 \]

\[ D_2 = X_1X_2 + X_1X_2Q_1' + X_2Q_2 \]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
15.32  
See answers to 14.23 for the state table.  
The four-state table is minimum.  
I.  \((S_0, S_1) \times 3 (S_0, S_2) (S_2, S_3) \times 3\)  
II.  \((S_0, S_1) (S_0, S_1, S_2) (S_2, S_3) (S_0, S_2, S_3)\)  
III.  \((S_0, S_1) \times 2 (S_2, S_3)\)

\[
\begin{array}{c|c|c|c|c|c|c}
Q_1Q_2 & Q'_1Q'_2 & Z \\
\hline
00 & 01 & 01 & 00 & 00 & 0 \\
01 & 01 & 11 & 00 & 01 & 0 \\
11 & 11 & 10 & 11 & 10 & 1 \\
10 & 11 & 10 & 00 & 10 & 1 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
Q_1 & \bar{Q}_1 & Q_2 & \bar{Q}_2 & Z \\
\hline
0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 \\
\end{array}
\]

\[
D_1 = X_1Q_1 + X_1X_2Q_1 + X_2Q_1 \\
D_2 = X_1X_2' + X_1Q_1' + X_2Q_1Q_2
\]

15.33

\[
\begin{array}{c|c|c|c|c|c|c}
A^* & B' & C' & W & A & B & C \\
\hline
00 & 0 1 1 0 & 0 & 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 \\
01 & 0 0 0 1 & 0 & 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 \\
11 & 1 1 0 1 & 1 & 1 1 1 & 1 1 1 1 & 1 1 1 1 & 1 1 1 1 \\
10 & 1 1 0 1 & 1 & 1 1 1 & 1 1 1 1 & 1 1 1 1 & 1 1 1 1 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
T_A & T_B & T_C \\
\hline
\hline
00 & 0 1 1 1 & 0 & 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 \\
01 & 1 0 0 0 & 0 & 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 \\
11 & 1 1 0 1 & 0 & 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 \\
10 & 1 1 0 1 & 0 & 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 & 0 0 0 0 \\
\end{array}
\]

\[
0 = 1 = 3 = 5
\]
Unit 15 Solutions

15.33
I. None
II. (4, 7) ✓ (6, 7) ✓ (2, 4) ✓ (2, 6) ✓

Assignment:
\[ S_0 = 000, S_2 = 100, S_4 = 111, S_6 = 110, S_7 = 101 \]

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W = 0 )</td>
<td>1</td>
<td>0 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W = 0 )</td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>

15.34

\[ T_A = 0; \quad T_B = W'A; \quad T_C = WB + AB'; \quad Z = W'AB'C' \]

15.35

By inspecting incoming arrows, we get:
\[ Q_0^* = D_0 = X'YQ_0 + YQ_1 + X'YQ_2 \]
\[ Q_1^* = D_1 = XYQ_0 + YQ_1 + YQ_2 \]
\[ Q_2^* = D_2 = XYQ_0 + X'YQ_1 + YQ_2 \]
\[ Z = XYQ_1 + XYQ_2 + X'YQ_2 = X'YQ_1 + YQ_2 \]

15.36

<table>
<thead>
<tr>
<th>( Q_2 )</th>
<th>( Q_1 )</th>
<th>( Q_0 )</th>
<th>( Clr, Ld, Cnt )</th>
<th>( X = 0 )</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>011</td>
<td>010</td>
<td>101 101</td>
<td>000</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>010</td>
<td>010</td>
<td>101 101</td>
<td>010</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>000</td>
<td>000</td>
<td>11-</td>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
<td>11-</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>101</td>
<td>101</td>
<td>11-</td>
<td>101</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>100</td>
<td>100</td>
<td>11-</td>
<td>110</td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>111</td>
<td>111</td>
<td>11-</td>
<td>111</td>
<td>1</td>
</tr>
</tbody>
</table>

15.37 (a)

By inspecting incoming arrows, we get:
\[ D_0 = Q_0^* = X \]
\[ D_1 = Q_0^* = X'Q_0 \]
\[ D_2 = Q_0^* = Q_0' \]
\[ D_3 = Q_0^* = Q_0' \]
\[ Z = XQ_0' \]

15.37 (b)

<table>
<thead>
<tr>
<th>( Q_0 )</th>
<th>( Q_1 )</th>
<th>( Q_0' )</th>
<th>( Z )</th>
<th>( X = 0 )</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>010</td>
<td>000</td>
<td>0 0</td>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>110</td>
<td>000</td>
<td>0 0</td>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>000</td>
<td>0 1</td>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>110</td>
<td>000</td>
<td>0 0</td>
<td>0 0</td>
<td>1</td>
</tr>
</tbody>
</table>

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
15.37 (c) For the counter, a better state assignment is \( A = 00, B = 01, C = 10 \) and \( D = 11 \).

<table>
<thead>
<tr>
<th>( Q_sQ_0 )</th>
<th>( s_s0 )</th>
<th>( Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01 11</td>
<td>0 0</td>
</tr>
<tr>
<td>01</td>
<td>01 11</td>
<td>0 0</td>
</tr>
<tr>
<td>11</td>
<td>00 11</td>
<td>0 0</td>
</tr>
<tr>
<td>10</td>
<td>01 11</td>
<td>0 1</td>
</tr>
</tbody>
</table>

\( s_s = X_s, s_0 = X + Q_1', Z = XQ_2Q_0' \)

\( P_3 = -, P_2 = -, P_1 = -, P_0 = - \)

15.37 (d) Another possibility is to duplicate state \( D \) and use 1110 and 1111 as state assignments for the two \( D \)'s.

<table>
<thead>
<tr>
<th>( Q_sQ_2Q_1Q_0 )</th>
<th>( s_s0 )</th>
<th>( Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>01 11</td>
<td>0 0</td>
</tr>
<tr>
<td>1000</td>
<td>01 11</td>
<td>0 0</td>
</tr>
<tr>
<td>1100</td>
<td>01 11</td>
<td>0 1</td>
</tr>
<tr>
<td>1110</td>
<td>01 11</td>
<td>0 0</td>
</tr>
<tr>
<td>1111</td>
<td>01 11</td>
<td>0 0</td>
</tr>
</tbody>
</table>

\( s_s = X_s, s_0 = X + Q_1', Z = XQ_2Q_1', S_{in} = 1 \)

\( P_3 = -, P_2 = -, P_1 = -, P_0 = - \)

15.38 (b)  The equations for \( T_1 \) and \( T_2 \) are the same as in Part (a).

<table>
<thead>
<tr>
<th>( Q_1Q_2 )</th>
<th>( Q_1^+ Q_2^+ )</th>
<th>( T_1T_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00 01</td>
<td>00 00</td>
</tr>
<tr>
<td>01</td>
<td>00 10</td>
<td>01 00</td>
</tr>
<tr>
<td>11</td>
<td>00 11</td>
<td>11 00</td>
</tr>
<tr>
<td>10</td>
<td>00 11</td>
<td>10 00</td>
</tr>
</tbody>
</table>

\( Q_1^+ = XQ_1 + XQ_2 = XQ_1 + XQ_2(Q_1 + Q_1') = (X + Q_1')(X' + Q_2' + Q_1)Q_1 \)

\( Q_2^+ = (XQ_1 + XQ_2')Q_1 + (X'Q_1 + XQ_2')Q_1' \)

\( T_1 = (X'Q_1 + XQ_2')Q_1 \)

\( Q_2^+ = XQ_1 + XQ_2' = XQ_1(Q_2 + Q_2') + XQ_2' = XQ_1Q_2 + XQ_2' \)

15.38 (c)  The equations for \( J_1, K_1, J_2 \) and \( K_2 \) are the same as in Part (c).

<table>
<thead>
<tr>
<th>( Q_1Q_2 )</th>
<th>( J_1K_1J_2K_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0, 0, 0, 0, 1</td>
</tr>
<tr>
<td>01</td>
<td>0, -1, 1, -1</td>
</tr>
<tr>
<td>11</td>
<td>-1, 0, -0, 1</td>
</tr>
<tr>
<td>10</td>
<td>-1, -1, 0, 0</td>
</tr>
</tbody>
</table>

\( Q_1^+ = J_1Q_1' + K_1Q_1 = Q_2Q_1' + Q_1Q_1 = Q_2Q_1' \)

\( T_1 = Q_1 + Q_2Q_1' = Q_1 + Q_2 \)

\( Q_2^+ = J_2K_2Q_2' = (X + Q_1')Q_2' + (1)Q_2 = X + Q_1' \)

<table>
<thead>
<tr>
<th>( Q_1Q_2 )</th>
<th>( J_1K_1J_2K_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0, 0, 0, 0, 1</td>
</tr>
<tr>
<td>01</td>
<td>0, -1, 1, -1</td>
</tr>
<tr>
<td>11</td>
<td>-1, 0, -0, 1</td>
</tr>
<tr>
<td>10</td>
<td>-1, -1, 0, 0</td>
</tr>
</tbody>
</table>

\( T_2 = Q_2 + (X + Q_1')Q_2' = Q_2 + X + Q_1' \)

\( Q_1^+ = J_1Q_1' + K_1Q_1 = Q_2Q_1' + Q_1Q_1 = Q_2Q_1' \)

\( T_1 = Q_1 + Q_2Q_1' = Q_1 + Q_2 \)

\( Q_2^+ = J_2K_2Q_2' = (X + Q_1')Q_2' + (1)Q_2 = X + Q_1' \)
### Unit 15 Solutions

#### 15.39 (b)

<table>
<thead>
<tr>
<th>$Q_1Q_2$</th>
<th>$J_1K_1, J_2K_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00, 11</td>
</tr>
<tr>
<td>01</td>
<td>10, 11</td>
</tr>
<tr>
<td>11</td>
<td>11, 01</td>
</tr>
<tr>
<td>10</td>
<td>01, 01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Q_1Q_2$</th>
<th>$Q_1^+Q_2^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01, 01</td>
</tr>
<tr>
<td>01</td>
<td>10, 10</td>
</tr>
<tr>
<td>11</td>
<td>00, 00</td>
</tr>
<tr>
<td>10</td>
<td>00, 01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Q_1Q_2$</th>
<th>$T_1T_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01, 01</td>
</tr>
<tr>
<td>01</td>
<td>11, 11</td>
</tr>
<tr>
<td>11</td>
<td>11, 11</td>
</tr>
<tr>
<td>10</td>
<td>10, 11</td>
</tr>
</tbody>
</table>

The equations for $T_1$ and $T_2$ are the same as in Part (a).

#### 15.39 (c)

$Q_1^+ = S_1 + R_1Q_1 = Q_2Q_1' + Q_1'Q_1$

so $S_1 = Q_2Q_1'$ and $R_1 = Q_1$

$Q_2^+ = S_2 + R_2Q_2 = (X + Q_1')Q_2' + (Q_2')'Q_2$

so $S_2 = (X + Q_1')Q_2'$ and $R_2 = Q_2$

#### 15.39 (d)

<table>
<thead>
<tr>
<th>$Q_1Q_2$</th>
<th>$S_1R_1, S_2R_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01, 01</td>
</tr>
<tr>
<td>01</td>
<td>10, 01</td>
</tr>
<tr>
<td>11</td>
<td>01, 01</td>
</tr>
<tr>
<td>10</td>
<td>01, 01</td>
</tr>
</tbody>
</table>

The equations for $S_1, R_1, S_2$ and $R_2$ are the same as in Part (c).

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
16.17 (a) The state meanings are given in the following table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>No 1’s have occurred</td>
</tr>
<tr>
<td>$S_1$</td>
<td>One 1 has occurred (an odd number &lt; 2)</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Two 1’s or an even number of 1’s &gt; 2 have occurred</td>
</tr>
<tr>
<td>$S_3$</td>
<td>An odd number of 1’s &gt; 2 has occurred.</td>
</tr>
</tbody>
</table>

16.17 (b)  

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_1$</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_2$</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_3$</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Note: Solution in FLD p. 729 uses state assignment $S_0 = 00$, $S_1 = 01$, $S_2 = 10$, $S_3 = 11$. 

16.17 (c) Since no 1’s have occurred, $a_i$ and $b_i$ are the same as $S_0$ or, $a_i = 0$; $b_i = 0$;

$\begin{align*}
    a_{i+1} &= x_i a'_i + x'_i a_i = x_i \\
    b_{i+1} &= b_i + x_i a_i = 0
\end{align*}$

first cell
Unit 16 Solutions
16.17 (d)

16.18 (a) The output becomes 1 whenever an even #0's or an even #1's (greater than 0) occurs.

The state meanings are given in the following table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>even #0's and even #1's received</td>
</tr>
<tr>
<td>$S_1$</td>
<td>even #0's and odd #1's received</td>
</tr>
<tr>
<td>$S_2$</td>
<td>odd #0's and even #1's received</td>
</tr>
<tr>
<td>$S_3$</td>
<td>odd #0's and odd #1's received</td>
</tr>
</tbody>
</table>

Guidelines: I: --
II: (1, 2)2x, (0, 3)2x

An assignment is

$$D_A = X'A' + XA$$
$$D_B = B'$$
$$Z = X'A + XAB + AB'$$
16.18 (b) The state meanings are given in the following table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>reset state</td>
</tr>
<tr>
<td>S1</td>
<td>even #0's and even #1's received</td>
</tr>
<tr>
<td>S2</td>
<td>odd #0's and even #1's received</td>
</tr>
<tr>
<td>S3</td>
<td>odd #0's and even #1's received</td>
</tr>
<tr>
<td>S4</td>
<td>even #0's and odd #1's received</td>
</tr>
<tr>
<td>S5</td>
<td>even #0's and odd #1's received</td>
</tr>
<tr>
<td>S6</td>
<td>odd #0's and odd #1's received</td>
</tr>
</tbody>
</table>

Guidelines:  I: (0, 1)2x, (2, 3)2x, (4, 5)2x  
             II: (2, 5)2x, (1, 6)4x, (3, 4)  

An assignment is

<table>
<thead>
<tr>
<th>ABC</th>
<th>010</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>S2</td>
</tr>
<tr>
<td>S1</td>
<td>S4</td>
</tr>
<tr>
<td>S2</td>
<td>S6</td>
</tr>
<tr>
<td>S3</td>
<td>S5</td>
</tr>
<tr>
<td>S4</td>
<td>X</td>
</tr>
<tr>
<td>S5</td>
<td>S6</td>
</tr>
</tbody>
</table>

16.18 (c) The state meanings are given in the following table:

<table>
<thead>
<tr>
<th>Present State</th>
<th>X = 0</th>
<th>1</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S6</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A'B'C'</th>
<th>X = 0</th>
<th>1</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>000</td>
<td>110 100</td>
<td>0</td>
</tr>
<tr>
<td>S1</td>
<td>001</td>
<td>110 100</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td>011</td>
<td>101 111</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>010</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>100</td>
<td>011 001</td>
<td>0</td>
</tr>
<tr>
<td>S4</td>
<td>101</td>
<td>011 001</td>
<td>1</td>
</tr>
<tr>
<td>S5</td>
<td>111</td>
<td>001 011</td>
<td>1</td>
</tr>
<tr>
<td>S6</td>
<td>110</td>
<td>001 011</td>
<td>0</td>
</tr>
</tbody>
</table>
Unit 16 Solutions

16.18 (c) (contd)

16.19 (a)

For assignment \( S_0 = 00, S_1 = 01, S_2 = 11: \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>X</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( A + B + )</th>
<th>( X = 0 )</th>
<th>( X = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 00 01</td>
<td>0 1</td>
<td></td>
</tr>
<tr>
<td>01 00 11</td>
<td>1 0</td>
<td></td>
</tr>
<tr>
<td>11 01 11</td>
<td>0 1</td>
<td></td>
</tr>
<tr>
<td>10 - - -</td>
<td>- -</td>
<td></td>
</tr>
</tbody>
</table>

16.19 (b)

<table>
<thead>
<tr>
<th>( A )</th>
<th>( B )</th>
<th>( X )</th>
<th>( J_A )</th>
<th>( K_A )</th>
<th>( J_B )</th>
<th>( K_B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>0</td>
<td>X X</td>
<td>X X</td>
<td>00</td>
<td>X X</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>1</td>
<td>X X</td>
<td>X X</td>
<td>01</td>
<td>X X</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>0</td>
<td>1 0</td>
<td>1 0</td>
<td>11</td>
<td>0 0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>X</td>
<td>X X</td>
<td>X X</td>
<td>10</td>
<td>X X</td>
</tr>
</tbody>
</table>

16.19 (c)

<table>
<thead>
<tr>
<th>( A )</th>
<th>( B )</th>
<th>( X )</th>
<th>( D )</th>
<th>( D \cdot Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>- 1</td>
<td>1</td>
<td>1</td>
<td>0 0</td>
<td></td>
</tr>
<tr>
<td>1 -</td>
<td>-</td>
<td>0</td>
<td>1 0</td>
<td></td>
</tr>
<tr>
<td>- -</td>
<td>1</td>
<td>0</td>
<td>1 0</td>
<td></td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 0 1</td>
<td>0</td>
<td>0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 - 1</td>
<td>0</td>
<td>0 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( D_a = X B \)
\( D_a = X + A \)
\( Z = X A' B + X B' + X A \)
\( Z = (X + B)(X + A')(X' + A + B') \)
16.20 (a) Inputs: \(X_2X_1X_0\) Outputs: ID

16.20 (b) 

\[D_0 = X_2'X_1'X_0 + X_2'X_1X_0'\]  
\[D_1 = (X_2'X_1' + X_2'X_0)(Q_0 + Q_2 + Q_4)\]  
\[D_2 = (X_2'X_1' + X_2'X_0)Q_1\]  
\[D_3 = (X_2'X_1' + X_2'X_0)Q_2\]  
\[D_4 = (X_2'X_1' + X_2'X_0Q_3\]  
\[D_5 = (X_2'X_1' + X_2'X_0Q_4\]  

16.20 (c) Using the assignment \(S_0 = 000, S_1 = 001, S_2 = 010, S_3 = 011, S_4 = 100, S_5 = 101, S_6 = 110, S_7 = 111:\)

\[D_0 = (X_2X_1X_0 + X_2X_1'X_0')(S_0 + S_1 + S_2 + S_3 + S_4 + S_5 + S_6)\]  
\[D_1 = (X_2'X_1 + X_2X_0)(S_1 + S_2 + S_3 + S_4 + S_6) + (X_2'X_1' + X_1X_0)(S_1)\]  
\[D_2 = (X_2'X_1 + X_2X_0)(S_2 + S_3 + S_4 + S_6) + (X_2'X_1' + X_2X_0)(S_1')\]  
\[D_3 = (X_2'X_1 + X_2X_0)(S_4 + S_6) + (X_2'X_1' + X_2X_0)(S_3')\]  
\[D_4 = (X_2'X_1 + X_2X_0)(S_1') + (X_2'X_1' + X_1X_0)(S_1')\]  
\[D_5 = (X_2'X_1 + X_2X_0)(S_4) + (X_2'X_1' + X_1X_0)(S_4)\]  
\[D_6 = (X_2'X_1 + X_2X_0)(S_6)\]  

\[I = S_5 + Q_2\]  
\[D = S_6 + Q_2Q_0' + Q_2Q_1'\]  

* \(S_7\) never occurs so 111 is a don’t care input combination.
16.21 (c) Using the assignment \(S_0 = 000, S_1 = 001, S_2 = 010, S_3 = 011, S_4 = 100\):
\[
D_2 = X_2X_0Q_1Q_0' + X_2X_0Q_1Q_0' + X_2X_0Q_2 + X_2X_1Q_0 \\
= X_2X_0Q_1Q_0' + X_2X_0Q_1Q_0' + X_2X_0Q_0 + X_2'X_1Q_0
\]
\[
D_1 = X_2X_0Q_2Q_1' + X_2X_2Q_2Q_1' + X_2X_1Q_0 + X_1X_0Q_0 + X_2'X_0Q_0
\]
\[
D_0 = X_2'X_1Q_0 + X_2'X_0Q_0Q_0 \\
D = X_2X_0Q_2 + X_2X_1Q_0
\]

16.22 (a) Inputs: \(XY\) Outputs: \(Z_2Z_1Z_0\)

16.22 (b) \(D_0 = 0\)
\[
D_1 = (XY' + XY)(Q_0 + Q_4 + Q_5 + Q_6) \\
D_2 = (XY' + XY)Q_1 \\
D_3 = (XY' + XY)(Q_2 + Q_3) \\
D_4 = (XY + XY)(Q_0 + Q_4 + Q_5) \\
D_5 = (XY + XY)(Q_2 + Q_3) \\
D_6 = (XY + XY)(Q_0 + Q_4 + Q_5)
\]

16.22 (c) Using the assignment \(S_0 = 000, S_1 = 001, S_2 = 010, S_3 = 011, S_4 = 100, S_5 = 101, S_6 = 110\):
\[
D_0 = (XY' + XY)(Q_0 + Q_4 + Q_5 + Q_6) \\
= (XY' + XY)Q_0 + Q_1 + Q_2 + Q_3 + (XY' + XY)Q_6 \\
= (XY' + XY)(Q_0' + Q_1 + Q_2 + Q_3) + (XY' + XY)Q_6 \\
D_1 = (XY' + XY)(Q_0' + Q_1 + Q_2 + Q_3) + (XY' + XY)Q_6 \\
= (XY' + XY)(Q_0' + Q_1 + Q_2 + Q_3 + Q_4 + Q_5) \\
D_2 = (XY + XY)(Q_0' + Q_1 + Q_2 + Q_3 + Q_4 + Q_5) \\
= (XY' + XY)Q_0 + Q_1 + Q_2 + Q_3 + Q_4 + Q_5 \\
Z_0 = Q_1 + Q_3 + Q_5 \\
Z_1 = Q_2 + Q_3 + Q_6 \\
Z_2 = Q_4 + Q_5 + Q_6
\]

16.23 (a) Inputs: \(XY\) Outputs: \(Z_2Z_1Z_0\)

16.23 (b) \(D_0 = 0\)
\[
D_1 = (XY' + XY)(Q_0 + Q_5 + Q_6) \\
D_2 = (XY' + XY)(Q_1 + Q_2) \\
D_3 = (XY + XY)(Q_0 + Q_1 + Q_2) \\
D_4 = (XY + XY)(Q_6 + Q_5) \\
D_5 = (XY + XY)(Q_0 + Q_1 + Q_2) \\
D_6 = (XY + XY)(Q_6 + Q_5)
\]

* \(S7\) never occurs so \(111\) is a don’t care input combination.
16.23 (c) Using the assignment \( S_0 = 000, S_1 = 001, S_2 = 101, \) 
\( S_3 = 011, S_4 = 100, S_5 = 101, S_6 = 110, S_7 = 111: \)
\[
\begin{align*}
D_2 &= Q_1 Q'_0 + X'Y Q'_1 + X'Y Q'_1' \\
D_1 &= X'Y + XY \\
D_0 &= 1 \\
Z &= Q + Q' + Q \\
Z_1 &= Q_1 Q'_0 + X'Y Q'_1 + X'Y Q'_1' \\
Z_2 &= X'Y + XY
\end{align*}
\]
\[
\begin{align*}
Z_0 &= X'Y Q'_0' + X'Y Q'_0 + X'Y Q'_2 + X Y Q_2 \\
&\quad + Q_2 Q_1 + X'Y Q'_1 + X Y Q_1
\end{align*}
\]

16.24 (b) \( D_0 = 0 \)
\[
\begin{align*}
D_1 &= (X'Y' + XY)(Q_0 + Q_4 + Q_5 + Q_6 + Q_7) \\
D_2 &= (X'Y' + XY)Q_1 \\
D_3 &= (X'Y' + XY)Q_2 + Q_3 \\
D_4 &= (X'Y' + XY)(Q_4 + Q_3) \\
D_5 &= (X'Y + XY)Q_6 \\
D_6 &= (X'Y + XY)Q_7 \\
Z_0 &= Q_1 + Q_3 + Q_6 + Q_7 \\
Z_1 &= Q_2 + Q_3 + Q_6 + Q_7 \\
Z_2 &= Q_4 + Q_5 + Q_6 + Q_7 \\
\end{align*}
\]

16.24 (c) Using the assignment \( S_0 = 000, S_1 = 001, S_2 = 010, \)
\( S_3 = 011, S_4 = 100, S_5 = 101, S_6 = 110, S_7 = 111: \)
\[
\begin{align*}
D_0 &= (X'Y' + XY)(S_0 + S_4 + S_5 + S_6 + S_7 + S_2 + S_3 + (X'Y + XY)(S_0 + S_1 + S_2 + S_3) \\
&\quad + (X'Y + XY)(S_0 + S_1 + S_2 + S_3 + S_6)) \\
&\quad + (X'Y' + XY)(Q_0' + Q_1 + Q_2 + Q_3) \\
D_1 &= (X'Y' + XY)(S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + S_7) \\
&\quad + (X'Y + XY)(Q_2' + Q_1 + Q_0) \\
D_2 &= (X'Y + XY)
\end{align*}
\]
\[
\begin{align*}
Z_0 &= S_1 + S_3 + S_5 + S_7 = Q_0 \\
Z_1 &= S_2 + S_3 + S_5 + S_7 = Q_1 \\
Z_2 &= S_4 + S_5 + S_6 + S_7 = Q_2 \\
\end{align*}
\]

16.25 (a) Inputs: XY Outputs: \( Z_2 Z_1 Z_0 \)

16.25 (b) \( D_0 = 0 \)
\[
\begin{align*}
D_1 &= (X'Y' + XY)(Q_0 + Q_5 + Q_6 + Q_7) \\
D_2 &= (X'Y' + XY)(Q_1 + Q_2) \\
D_3 &= (X'Y' + XY)(Q_2 + Q_3) \\
D_4 &= (X'Y' + XY)(Q_4 + Q_3) \\
D_5 &= (X'Y + XY)(Q_6) \\
D_6 &= (X'Y + XY)(Q_7) \\
Z_0 &= Q_0 + (X'Y' + XY)(Q_2 + Q_3) \\
&\quad + (X'Y + XY)(Q_1 + Q_2) \\
Z_1 &= Q_1 + Q_2 + (X'Y + XY)(Q_0 + Q_7) \\
Z_2 &= (X'Y + XY)
\end{align*}
\]
Unit 16 Solutions

16.25 (c) Using the assignment \( S_0 = 000, S_1 = 001, S_2 = 100, S_3 = 111, S_5 = 0011, S_6 = 010, S_7 = 111 \):

- \( D_2 = X'YQ_1' + XYQ_1' + Q_1Q_0 \)
- \( D_1 = X'Y + XY' \)
- \( D_0 = Q_1' + XY + Q_2' \)
- \( Z_2 = X'Y + XY' \)
- \( Z_1 = X'YQ_1' + XYQ_1' + Q_1Q_0 + XYQ_2 + XYQ_2' \)
- \( Z_0 = Q_0' + XYQ_1' + XYQ_1' + XYQ_1 + Q_2Q_1' \)

16.26 (b)

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>Staying on first floor</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>Moving from first to second floor</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>Staying on second floor</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>Moving from second to first floor</td>
</tr>
</tbody>
</table>

16.26 (c) With the state assignment \( S_0 = 00, S_1 = 01, S_2 = 10, S_3 = 11 \), we have:

- \( D_1 = FS_2Q_1Q_2 + FS_1Q_1 + Q_2Q_2' \)
- \( D_2 = FS_2Q_1Q_2 + FS_1Q_1 + N_1N_2DCQ_1Q_2 + N_1N_2DCQ_1Q_2' \)
- \( R_1 = FS_2Q_1Q_2 + N_1Q_1Q_2' \)
- \( R_2 = FS_2Q_1Q_2 + N_2Q_1Q_2' \)
- \( DOWN = FS_1Q_1Q_2 + N_1N_2DCQ_1Q_2' \)
- \( UP = FS_2Q_1Q_2 + N_1N_2DCQ_1Q_2' \)
- \( DO = FS_2Q_1Q_2 + FS_1Q_1Q_2 + N_1Q_1Q_2 + N_2Q_1Q_2' \)

16.27 (a)

Outputs: \( LC, LB, LA, RA, RB, RC \)
16.27 (b) First, assign $LC = Q_1$, $LB = Q_2$, $LA = Q_3$, $RA = Q_4$, $RB = Q_5$, $RC = Q_6$. So $S_0 = 000000$, $S_1 = 001000$, $S_2 = 011000$, etc.

This state machine has too many state variables to use Karnaugh maps. Instead, we will write down equations for each flip-flop by inspection.

First consider $Q_1$: $Q_1 = 1$ in states $S_0$ or $S_1$ only.

- $S_0$ is reached whenever $H = 1$ and we are not already in $S_1$: $HQ_1'Q_2'Q_3'Q_4'Q_5'Q_6'$. But $S_1$ is the only state in which both $Q_1 = 1$ and $Q_4 = 1$, so assuming we are always in a valid state, we can use $H(Q_1Q_4)' = HQ_1' + HQ_4'$. Note: Any combination of one left light and one right light will also work, i.e. $HQ_1' + HQ_4'$.

- $S_2$ is reached whenever we are in $S_1$ and $L = 1$ while $H = 0$: $LHQ_1'Q_2'Q_3'Q_4'Q_5'Q_6'$. But $Q_3 = 1$ whenever $Q_2 = 1$, and $Q_4 = Q_5 = Q_6 = 0$ whenever $Q_1 = 0$. So we can use $LHQ_1'Q_2'$.

- So $D_1 = LHQ_1'Q_2' + HQ_1' + HQ_4' = LQ_2'Q_4 + HQ_1' + HQ_4'$ (using $X + X'Y = X + Y$).

Similarly $Q_2 = 1$ in states $S_0$, $S_2$, and $S_3$ only.

- $S_3$ and $S_4$ are reached whenever we are in $S_0$ or $S_2$ and $L = 1$ while $H = 0$.

$LHQ_1'Q_2'Q_3'Q_4' + LHQ_1'Q_2'Q_3'Q_4' = LHQ_1'Q_2'Q_3'Q_4'Q_5'Q_6'$

But again, $Q_1 = Q_2 = Q_3 = 0$ whenever $Q_1 = 0$, so $D_3 = LQ_3'Q_4' + HQ_1' + HQ_4'$. We can also get by inspection: $D_3 = LQ_3'Q_4' + HQ_1' + HQ_4'$. $D_4 = RQ_3'Q_6' + HQ_3' + HQ_4'$.

16.27 (c)

<table>
<thead>
<tr>
<th>State</th>
<th>$LRH = 000$</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_7$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_1$</td>
</tr>
</tbody>
</table>

I. $(S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7)$ for $S_1$ in $LRH = 001, 011, 101$

$(S_0, S_1, S_2, S_3, S_4)$ for $S_2$ in $LRH = 010$

$(S_0, S_1, S_2, S_3, S_4)$ for $S_3$ in $LRH = 100$

II. Every state matches $S_0$ and $S_2$. But $S_0$ and $S_2$ match the best, so $(S_0, S_2, S_5, S_7, S_1, S_3, S_4)$ etc.

From LogicAid:

$D_1 = HQ_1' + RQ_1Q_2Q_3' + HQ_1' + LQ_1Q_3'Q_4' + HQ_1' + RQ_1Q_2Q_3'$

$D_2 = RHQ_1Q_2Q_3' + RHQ_1Q_4 + HQ_1'Q_4'$

$D_3 = LHQ_1'Q_2Q_3' + LHQ_1'Q_3'Q_4 + RQ_1Q_2Q_3'$

$D_4 = RQ_3'Q_6' + HQ_3' + HQ_4'$

$LC = Q_1'Q_2'$

$LB = Q_1'Q_2' + Q_1'Q_3'$

$LA = Q_1'Q_3' + Q_1'Q_4' + Q_1'Q_2Q_3'$

$RC = Q_1'Q_2Q_3' + Q_1'Q_2Q_4'$

$RB = Q_1'Q_2Q_3' + Q_1'Q_2Q_3'$

$RA = Q_1'Q_3' + Q_2Q_3'$

Other minimum solutions can be found for $D_2$ and $D_3$ with this assignment.
Note: This state graph assumes that only one of the buttons ST, PL, RE, and FF can be pressed at any given time. The graph is incompletely specified and must be augmented before using LogicAid. For example, the arc from REW to PLAY should be labeled PL ST' FF'.

\[ D_1 = ST' FF PS Q_1 Q_2 Q_3 + ST' RE PL Q_1' Q_2 Q_3 + ST' M Q_1 \]
\[ D_2 = ST' FF Q_1 Q_2 Q_3' + ST' RE Q_1' Q_2 Q_3' + ST' RE' PL Q_1 Q_2 + ST' FF' PL Q_1 Q_2' \]
\[ D_3 = ST' FF Q_1 Q_2 Q_3' + ST' RE FF' Q_1' Q_2' + ST' FF' PL Q_1' Q_2' + ST' FF' PL Q_1 Q_2' + ST' RE' FF' Q_1' Q_2' + ST' RE' FF' Q_1 Q_2' \]
\[ P = Q_1 Q_2 Q_3' \]
\[ R = Q_1 Q_2' + Q_1' Q_2' \]
\[ F = Q_1 Q_3' + Q_1' Q_3' \]
16.29 (a)

16.29 (b)

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>X = 0</th>
<th>X = 1</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_0</td>
<td>S_0 S_1</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>S_1</td>
<td>S_2 S_3</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>S_2</td>
<td>S_0 S_2</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>S_3</td>
<td>S_3 S_3</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>S_4</td>
<td>S_4 S_2</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>S_5</td>
<td>S_5 S_6</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>S_6</td>
<td>S_6 S_6</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

16.29 (c)

\[ a_{i+1} = x_i b_i c_i + a_i b_i + x_i b_i c_i' + x_i a_i \]

16.30 (a)

Z = a_n b_n
Unit 16 Solutions

16.30 (b)

<table>
<thead>
<tr>
<th>State</th>
<th>$x_i = 0$</th>
<th>$x_i = 1$</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_2$</td>
<td>$S_1$</td>
<td>1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_3$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$a_i b_i$</th>
<th>$a_{i+1} b_{i+1}$</th>
<th>$x_i = 0$</th>
<th>$x_i = 1$</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>10 01</td>
<td>01 11</td>
<td>01 11</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>10 11</td>
<td>10 11</td>
<td>10 11</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>01 01</td>
<td>01 01</td>
<td>01 01</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ a_{i+1} = (x_i + a_i) (x_i' + b_i') \]
\[ b_{i+1} = (x_i + b_i) (x_i' + a_i') \]

16.30 (c) $a_1 = b_1 = 0$
\[ a_2 = (x_1 + 0)(x_1' + 1) = x_1 \]
\[ b_2 = (x_1 + 1)(x_1' + 0) = x_1 \]

16.30 (d)
Unit 17 Problem Solutions

17.1  See FLD p. 731 for solution.

17.3 (a, b)

17.4  See FLD p. 733-734 for solution.

17.6  See FLD p. 734-735 for solutions.

17.7 (a) See FLD p. 736 for solution.

17.8  See FLD p. 738 for solution.
17.9 library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
    use IEEE.STD_LOGIC_ARITH.ALL;
    use IEEE.STD_LOGIC_UNSIGNED.ALL;
    entity srff
    port (clk, s, r : in std_logic;
          q, qn : out std_logic);
    end srff;
    architecture Behavioral of srff is
    signal qint : std_logic:='0';
    begin
    q <= qint;
    qn <= not qint;
    process(clk)
    begin
    if clk'event and clk='1' then
    if (not s and r)='1' then qint <= '0';
    elsif (s and not r)='1' then qint<='1';
    elsif (s and r)='1' then qint<='X'; end if;
    end if;
    end process;
    end Behavioral;
end

17.10 library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
    use IEEE.STD_LOGIC_ARITH.ALL;
    use IEEE.STD_LOGIC_UNSIGNED.ALL;
    -- D-G Latch
    entity dglatch is
    port (d, g : in bit;
          q : out bit);
    end dglatch;
    architecture Behavioral of dglatch is
    begin
    process(g, d)
    begin
    if g='1' then q <= d; end if;
    end process;
    end Behavioral;
    -- D flip flop using D-G latches
    library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
    use IEEE.STD_LOGIC_ARITH.ALL;
    use IEEE.STD_LOGIC_UNSIGNED.ALL;
    entity dff is
    port (d, clk : in bit;
          q : out bit);
    end dff;
    architecture Behavioral of dff is
    component dglatch is
    port (d, g : in bit;
          q : out bit);
    end component;
    signal p, clkn : bit;
    begin
    clkn <= not clk;
    dg1 : dglatch port map(d, clkn, p);
    dg2 : dglatch port map(p, clk, q);
    end Behavioral;
end

17.11 A rising edge triggered D-CE flip flop with asynchronous clear and preset.

17.12

17.12
17.13  library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity encoder is
  port (y0, y1, y2, y3 : in bit;
        a, b, c : out bit);
end encoder;
architecture Behavioral of encoder is
begin
  process(y0, y1, y2, y3)
  begin
    if y3='1' then
      a <= '1'; b <= '1'; c <= '1';
      -- y3 has highest priority
    elsif y2='1' then
      a <= '1'; b <= '0'; c <= '1';
    elsif y1='1' then
      a <= '0'; b <= '1'; c <= '1';
    elsif y0='1' then
      a <= '0'; b <= '0'; c <= '1';
    else
      a <= '0'; b <= '0'; c <= '0';
    end if;
  end process;
end Behavioral;

17.14  library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity comparator is
  port (a, b : in std_logic_vector(3 downto 0);
        agb, alb, aeb : out std_logic);
end comparator;
architecture Behavioral of comparator is
begin
  process(a, b)
  begin
    if a > b then
      agb <= '1'; alb <= '0'; aeb <= '0';
    elsif a < b then
      agb <= '0'; alb <= '1'; aeb <= '0';
    else
      agb <= '0'; alb <= '0'; aeb <= '1';
    end if;
  end process;
end Behavioral;

17.15  library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity super is
  port (a: in std_logic_vector(2 downto 0);
        d : in std_logic_vector(5 downto 0);
        rsi, lsi, clk : in std_logic;
        q : out std_logic_vector(5 downto 0));
end super;
architecture Behavioral of super is
signal qint: std_logic_vector(5 downto 0);
begin
  q <= qint;
  process(clk)
  begin
    if clk' event and clk='1' then
      case a is
        when "111"=> qint <= d;
        when "110"=> qint <= qint+1;
        when "101"=> qint <= "000000";
        when "100"=> qint <= "111111";
        when "011"=> qint <= "000000";
        when "010"=> qint <= rsi&qint(5 downto 1);
        when "001"=> qint <= qint(4 downto 0)&lsi;
        when others=> NULL;
      end case;
    end if;
  end process;
end Behavioral;

17.16  library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity bcd_seven is
  port (bcd : in bit_vector(3 downto 0);
        seven : out bit_vector(7 downto 1));
end bcd_seven;
architecture Behavioral of bcd_seven is
begin
  process(bcd)
  begin
    case bcd is
      when "0000"=> seven <= "01111111";
      when "0001"=> seven <= "00001101";
      when "0010"=> seven <= "10110110";
      when "0011"=> seven <= "10011111";
      when "0100"=> seven <= "11001101";
      when "0101"=> seven <= "11011011";
      when "0110"=> seven <= "11111111";
      when "0111"=> seven <= "00001111";
      when "1000"=> seven <= "11111111";
      when "1001"=> seven <= "11011111";
    end case;
  end process;
end Behavioral;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Mealy_XOR is
  Port (CLK, clr, x : in std_logic;
        z : out std_logic);
end Mealy_XOR;

architecture df1 of Mealy_XOR is
signal q, d: std_logic;
begin
  z <= x XOR q after 10ns;
  d <= x;
  process (CLK, clr)
  begin
    if clr = '0' then
      q <= '0' after 10ns;
    elsif CLK'event and CLK = '1' then
      q <= d after 10ns;
    end if;
  end process;
end df1;

architecture df1 of Moore_XOR is
signal Q1, Q2, D1, D2: std_logic;
begin
  Z <= Q1 XOR Q2 after 10ns;
  D1 <= X;
  D2 <= Q1;
  process (CLK, clr)
  begin
    if clr = '0' then
      Q1 <= '0' after 10ns;
      Q2 <= '0' after 10ns;
    elsif CLK'event and CLK = '1' then
      Q1 <= D1 after 10ns;
      Q2 <= D2 after 10ns;
    end if;
  end process;
end df1;
17.17 (e) The Mealy model output is valid before the positive clock edge while the corresponding Moore model output becomes valid after the clock edge. Also, the Mealy output is not valid after the clock edge until the input has changed to its next value. The Mealy model does not have an output corresponding to the Moore output prior to the first clock edge.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Current</th>
<th>0ns</th>
<th>40ns</th>
<th>80ns</th>
<th>120ns</th>
<th>160ns</th>
<th>200ns</th>
<th>240ns</th>
<th>280ns</th>
<th>320ns</th>
<th>360ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>z</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

17.18 (a) \[ Z_0 = Q_0 Q'_1 Q'_3' \text{ or } Q'_1 Q'_2 Q'_3' \]

\[ Z_1 = Q_0 Q_1 \]
\[ Z_2 = Q'_0 Q'_1 Q'_2' \text{ or } Q'_0 Q'_2 Q'_3' \]
\[ Z_3 = Q_1 Q_2 \]
\[ Z_4 = Q'_1 Q'_2 Q'_3' \text{ or } Q'_0 Q'_1 Q'_3' \]
\[ Z_5 = Q_2 Q_3 \]
\[ Z_6 = Q'_0 Q'_2 Q'_3' \text{ or } Q'_0 Q'_1 Q'_2' \]
\[ Z_7 = Q_0 Q_3 \]

17.18 (b) \[ D_0 = Q'_1 Q'_2' \]
\[ D_1 = Q'_2 Q'_3' \]
\[ D_2 = Q'_0 Q'_3' \]
\[ D_3 = Q'_0 Q'_1' \]

17.18 (c) \[ CE_0 = Q'_2' \]
\[ D_0 = Q'_1' \text{ or } CE_0 = Q'_1' \]
\[ D_0 = Q'_2' \]
\[ CE_1 = Q'_3' \]
\[ D_1 = Q'_2' \text{ or } CE_1 = Q'_2' \]
\[ D_1 = Q'_3' \]
\[ CE_2 = Q'_3' \]
\[ D_2 = Q'_0' \text{ or } CE_2 = Q'_0' \]
\[ D_2 = Q'_3' \]
\[ CE_3 = Q'_1' \]
\[ D_3 = Q'_0' \text{ or } CE_3 = Q'_0' \]
\[ D_3 = Q'_1' \]

17.18 (d) \[ \text{library IEEE;} \]
\[ \text{use IEEE.STD_LOGIC_1164.ALL;} \]
\[ \text{entity mod8_counter is} \]
\[ \text{port} (\text{CLK, ClrN : in std_logic;}) \]
\[ \text{Z0, Z1, Z2, Z3, Z4, Z5, Z6, Z7 : out std_logic;}) \]
\[ \text{end mod8_counter;} \]

\[ \text{architecture bhvr of mod8_counter is} \]
\[ \text{signal Q, Q_plus: std_logic_vector(0 to 3);} \]
\[ \begin{align*}
\text{begin} \\
\text{cmb_lgc: process(Q)} \\
\text{begin} \\
\text{Z0 <= '0'; Z1 <= '0'; Z2 <= '0'; Z3 <= '0';} \\
\text{Z4 <= '0'; Z5 <= '0'; Z6 <= '0'; Z7 <= '0';} \\
\text{case Q is} \\
\text{when "1000" =>} \\
\text{Z0 <= '1';} \\
\text{Q_plus <= "1100";} \\
\text{when "1100" =>} \\
\text{Z1 <= '1';} \\
\text{Q_plus <= "0100";} \\
\text{when "0100" =>} \\
\text{Z2 <= '1';} \\
\text{Q_plus <= "0110";} \\
\text{when "0110" =>} \\
\text{Z3 <= '1';} \\
\text{Q_plus <= "0010";} \\
\text{when "0010" =>} \\
\text{Z4 <= '1';} \\
\text{Q_plus <= "0011";} \\
\text{when "0011" =>} \\
\text{Z5 <= '1';} \\
\text{Q_plus <= "0001";} \\
\text{when "0001" =>} \\
\text{Z6 <= '1';} \\
\text{Q_plus <= "1001";} \\
\text{when "1001" =>} \\
\text{Z7 <= '1';} \\
\text{Q_plus <= "1000";} \\
\text{when others =>} \\
\text{Q_plus <= "XXXX";} \\
\text{end case;} \\
\text{end process cmb_lgc;} \\
\end{align*} \]

17.18 (d) \[ \text{stt_trnstn: process(CLK,ClrN)} \]
\[ \text{(contd)} \]
\[ \text{begin} \\
\text{if ClrN = '0' then} \\
\text{Q <= "1000";} \\
\text{elsif Rising_Edge (CLK) then} \\
\text{Q <= Q_plus;} \\
\text{end if;} \\
\text{end process stt_trnstn;} \\
\text{end bhvr;} \]
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mod8_counter is
port (CLK, ClrN : in std_logic;
    Z0, Z1, Z2, Z3, Z4, Z5, Z6, Z7 : out std_logic);
end mod8_counter;
architecture df1 of mod8_counter is
signal Q, D : std_logic_vector(0 to 3);
begincmb_lgc: process(Q)
begin
    Z0 <= Q(0) and not Q(1) and not Q(3);
    Z1 <= Q(0) and Q(1);
    Z2 <= not Q(0) and Q(1) and not Q(2);
    Z3 <= Q(1) and Q(2);
    Z4 <= not Q(1) and Q(2) and not Q(3);
    Z5 <= Q(2) and Q(3);
    Z6 <= not Q(0) and not Q(2) and Q(3);
    Z7 <= Q(0) and Q(3);
    D(0) <= not Q(1) and not Q(2);
    D(1) <= not Q(2) and not Q(3);
    D(2) <= not Q(0) and not Q(3);
    D(3) <= not Q(0) and not Q(1);
end process cmb_lgc;
stt_trnstn: process(CLK,ClrN)
begin
    if ClrN = '0' then
        Q <= "1000";
    elsif Rising_edge (CLK) then
        if CE(0) = '1' then
            Q(0) <= D(0);
        end if;
        if CE(1) = '1' then
            Q(1) <= D(1);
        end if;
        if CE(2) = '1' then
            Q(2) <= D(2);
        end if;
        if CE(3) = '1' then
            Q(3) <= D(3);
        end if;
    end if;
end process sttt_trnstn;
end df1;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mod8_counter is
port (CLK, ClrN : in std_logic;
    Z0, Z1, Z2, Z3, Z4, Z5, Z6, Z7 : out std_logic);
end mod8_counter;
architecture df2 of mod8_counter is
signal Q, CE, D : std_logic_vector(0 to 3);
begincmb_lgc: process(Q)
begin
    Z0 <= Q(0) and not Q(1) and not Q(3);
    Z1 <= Q(0) and Q(1);
    Z2 <= not Q(0) and Q(1) and not Q(2);
    Z3 <= Q(1) and Q(2);
    Z4 <= not Q(1) and Q(2) and not Q(3);
    Z5 <= Q(2) and Q(3);
    Z6 <= not Q(0) and not Q(2) and Q(3);
    Z7 <= Q(0) and Q(3);
    CE(0) <= not Q(2); D(0) <= not Q(1);
    CE(1) <= not Q(3); D(1) <= not Q(2);
    CE(2) <= not Q(0); D(2) <= not Q(3);
    CE(3) <= not Q(1); D(3) <= not Q(0);
end process cmb_lgc;
stt_trnstn: process(CLK,ClrN)
begin
    if ClrN = '0' then
        Q <= "1000";
    elsif Rising_edge (CLK) then
        if CE(0) = '1' then
            Q(0) <= D(0);
        end if;
        if CE(1) = '1' then
            Q(1) <= D(1);
        end if;
        if CE(2) = '1' then
            Q(2) <= D(2);
        end if;
        if CE(3) = '1' then
            Q(3) <= D(3);
        end if;
    end if;
end process sttt_trnstn;
end df2;

<table>
<thead>
<tr>
<th>Signal</th>
<th>Current</th>
<th>0ns</th>
<th>40ns</th>
<th>80ns</th>
<th>120ns</th>
<th>160ns</th>
<th>200ns</th>
<th>240ns</th>
<th>280ns</th>
<th>320ns</th>
<th>360ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClrN</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z0</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z1</td>
<td>'1'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z2</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z3</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z4</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z5</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z6</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z7</td>
<td>'0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>1100</td>
<td>8</td>
<td>C</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>0100</td>
<td>C</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>C</td>
<td>4</td>
</tr>
</tbody>
</table>
17.18 (f) wave form

<table>
<thead>
<tr>
<th>Signal</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK</td>
<td>'1'</td>
</tr>
<tr>
<td>ClrN</td>
<td>'1'</td>
</tr>
<tr>
<td>Z0</td>
<td>'0'</td>
</tr>
<tr>
<td>Z1</td>
<td>'1'</td>
</tr>
<tr>
<td>Z2</td>
<td>'0'</td>
</tr>
<tr>
<td>Z3</td>
<td>'0'</td>
</tr>
<tr>
<td>Z4</td>
<td>'0'</td>
</tr>
<tr>
<td>Z5</td>
<td>'0'</td>
</tr>
<tr>
<td>Z6</td>
<td>'0'</td>
</tr>
<tr>
<td>Z7</td>
<td>'0'</td>
</tr>
</tbody>
</table>

17.19 (a) 

\[
\begin{align*}
Z_0 &= Q_0 Q_1' Q_3' 	ext{ or } Q_1' Q_2' Q_3' \\
Z_1 &= Q_0 Q_2' \\
Z_2 &= Q_0 Q_1 Q_2 	ext{ or } Q_0 Q_2 Q_3' \\
Z_3 &= Q_0' Q_1 \\
Z_4 &= Q_1' Q_2 Q_3' 	ext{ or } Q_0' Q_1' Q_3' \\
Z_5 &= Q_0' Q_3 \\
Z_6 &= Q_0 Q_2 Q_3 	ext{ or } Q_0 Q_1' Q_2' \\
Z_7 &= Q_2' Q_3
\end{align*}
\]

17.19 (b) 

\[
\begin{align*}
D_0 &= Q_3 + Q_2' \\
D_1 &= Q_6 Q_3' \\
D_2 &= Q_0' + Q_1 \\
D_3 &= Q_1' Q_2
\end{align*}
\]

17.19 (c) 

\[
\begin{align*}
C E_0 &= Q_2' , D_0 = Q_1' \text{ or } C E_0 = Q_1' , D_0 = Q_2' \\
C E_1 &= Q_3' , D_1 = Q_2' \text{ or } C E_1 = Q_2' , D_1 = Q_3' \\
C E_2 &= Q_3' , D_2 = Q_0' \text{ or } C E_2 = Q_0' , D_2 = Q_3' \\
C E_3 &= Q_1' , D_3 = Q_0' \text{ or } C E_3 = Q_0' , D_3 = Q_1'
\end{align*}
\]

17.19 (d) 

\[
\begin{align*}
D_0 &= Q_3 + Q_2' \\
D_1 &= Q_0 Q_3' \\
D_2 &= Q_0' + Q_1 \\
D_3 &= Q_1' Q_2
\end{align*}
\]

17.19 (e) 

\[
\begin{align*}
D_0 &= Q_3 + Q_2' \\
D_1 &= Q_0 Q_3' \\
D_2 &= Q_0' + Q_1 \\
D_3 &= Q_1' Q_2
\end{align*}
\]

17.19 (f) library IEEE;

\[
\begin{align*}
\text{use IEEE.STD_LOGIC_1164.ALL;}
\end{align*}
\]

\[
\begin{align*}
\text{entity mod8_counter2 is}
\end{align*}
\]

\[
\begin{align*}
\text{port (CLK, ClrN : in std_logic;}
\end{align*}
\]

\[
\begin{align*}
Z_0, Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7 : \text{out std_logic);}
\end{align*}
\]

\[
\begin{align*}
\text{end mod8_counter2;}
\end{align*}
\]

\[
\begin{align*}
\text{architecture bhvr of mod8_counter2 is}
\end{align*}
\]

\[
\begin{align*}
\text{signal Q, Q_plus: std_logic_vector(0 to 3);}
\end{align*}
\]

\[
\begin{align*}
\text{begin}
\end{align*}
\]

\[
\begin{align*}
cmb_lgc: \text{process(Q)}
\end{align*}
\]

\[
\begin{align*}
\begin{align*}
\text{begin}
\end{align*}
\end{align*}
\]

\[
\begin{align*}
\text{begin if ClrN = '0' then}
\end{align*}
\]

\[
\begin{align*}
\text{Q <= "1000";}
\end{align*}
\]

\[
\begin{align*}
\text{elsif Rising_Edge (CLK) then}
\end{align*}
\]

\[
\begin{align*}
\text{Q <= Q_plus;}
\end{align*}
\]

\[
\begin{align*}
\text{end if;}
\end{align*}
\]

\[
\begin{align*}
\text{end process cmb_lgc;}
\end{align*}
\]

\[
\begin{align*}
\text{stt_trnstn: \text{process(CLK,ClrN)}}
\end{align*}
\]

\[
\begin{align*}
\begin{align*}
\text{begin}
\end{align*}
\end{align*}
\]

\[
\begin{align*}
\text{if ClrN = '0' then}
\end{align*}
\]

\[
\begin{align*}
\text{Q <= "1000";}
\end{align*}
\]

\[
\begin{align*}
\text{elsif Rising_Edge (CLK) then}
\end{align*}
\]

\[
\begin{align*}
\text{Q <= Q_plus;}
\end{align*}
\]

\[
\begin{align*}
\text{end if;}
\end{align*}
\]

\[
\begin{align*}
\text{end process stt_trnstn;}
\end{align*}
\]

\[
\begin{align*}
\text{end bhvr;}
\end{align*}
\]
Unit 17 Solutions

17.19 (e) library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
entity mod8_counter2 is
    port (CLK, ClrN : in std_logic;
          Z0, Z1, Z2, Z3, Z4, Z5, Z6, Z7 : out std_logic);
end mod8_counter2;
architecture df1 of mod8_counter2 is
    signal Q, D : std_logic_vector(0 to 3);
begin
    cmb_lgc: process(Q)
    begin
        Z0 <= Q(0) and not Q(1) and not Q(3);
        Z1 <= Q(1) and not Q(2);
        Z2 <= Q(0) and Q(1) and Q(2);
        Z3 <= not Q(0) and Q(1);
        Z4 <= not Q(1) and Q(2) and not Q(3);
        Z5 <= not Q(0) and Q(3);
        Z6 <= Q(0) and Q(2) and Q(3);
        Z7 <= not Q(2) and Q(3);
        D(0) <= Q(3) or not Q(2);
        D(1) <= Q(0) and not Q(3);
        D(2) <= not Q(0) or Q(1);
        D(3) <= not Q(1) and Q(2);
    end process cmb_lgc;
    stt_trnstn: process(CLK,ClrN)
    begin
        if ClrN = '0' then
            Q <= "1000";
        elsif Rising_Edge (CLK) then
            Q <= D;
        end if;
    end process stt_trnstn;
end df1;

17.19 (f) library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
entity mod8_counter2 is
    port (CLK, ClrN : in std_logic;
          Z0, Z1, Z2, Z3, Z4, Z5, Z6, Z7 : out std_logic);
end mod8_counter2;
architecture df2 of mod8_counter2 is
    signal Q, CE, D : std_logic_vector(0 to 3);
begin
    cmb_lgc: process(Q)
    begin
        Z0 <= Q(0) and not Q(1) and not Q(3);
        Z1 <= Q(1) and not Q(2);
        Z2 <= Q(0) and Q(1) and Q(2);
        Z3 <= not Q(0) and Q(1);
        Z4 <= not Q(1) and Q(2) and not Q(3);
        Z5 <= not Q(0) and Q(3);
        Z6 <= Q(0) and Q(2) and Q(3);
        Z7 <= not Q(2) and Q(3);
        CE(0) <= Q(2); D(0) <= Q(3);
        CE(1) <= not Q(3); D(1) <= Q(0);
        CE(2) <= Q(0); D(2) <= Q(1);
        CE(3) <= not Q(1); D(3) <= Q(2);
    end process cmb_lgc;
    stt_trnstn: process(CLK,ClrN)
    begin
        if ClrN = '0' then
            Q <= "1000";
        elsif Rising_Edge (CLK) then
            if CE(0) = '1' then Q(0) <= D(0); end if;
            if CE(1) = '1' then Q(1) <= D(1); end if;
            if CE(2) = '1' then Q(2) <= D(2); end if;
            if CE(3) = '1' then Q(3) <= D(3); end if;
        end if;
    end process stt_trnstn;
end df2;

17.19 (e) library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
entity mod8_counter2 is
    port (CLK, ClrN : in std_logic;
          Z0, Z1, Z2, Z3, Z4, Z5, Z6, Z7 : out std_logic);
end mod8_counter2;
architecture df1 of mod8_counter2 is
    signal Q, D : std_logic_vector(0 to 3);
begin
    cmb_lgc: process(Q)
    begin
        Z0 <= Q(0) and not Q(1) and not Q(3);
        Z1 <= Q(1) and not Q(2);
        Z2 <= Q(0) and Q(1) and Q(2);
        Z3 <= not Q(0) and Q(1);
        Z4 <= not Q(1) and Q(2) and not Q(3);
        Z5 <= not Q(0) and Q(3);
        Z6 <= Q(0) and Q(2) and Q(3);
        Z7 <= not Q(2) and Q(3);
        D(0) <= Q(3) or not Q(2);
        D(1) <= Q(0) and not Q(3);
        D(2) <= not Q(0) or Q(1);
        D(3) <= not Q(1) and Q(2);
    end process cmb_lgc;
    stt_trnstn: process(CLK,ClrN)
    begin
        if ClrN = '0' then
            Q <= "1000";
        elsif Rising_Edge (CLK) then
            if CE(0) = '1' then Q(0) <= D(0); end if;
            if CE(1) = '1' then Q(1) <= D(1); end if;
            if CE(2) = '1' then Q(2) <= D(2); end if;
            if CE(3) = '1' then Q(3) <= D(3); end if;
        end if;
    end process stt_trnstn;
end df1;
17.20(c) library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mag_comp_1bit is
port (a, b, gi, ei : in std_logic ;
go, eo : out std_logic);
end mag_comp_1bit;
architecture compdf of mag_comp_1bit is
begin
go <= gi or (ei and a and b');
eo <= ei and ((a AND b) OR (a' AND b'))
end compdf;

17.20(d) library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mag_comp_1bit is
port (a, b, gi, ei : in std_logic ;
go, eo : out std_logic);
end mag_comp_1bit;
architecture compdf of mag_comp_1bit is
begin
go <= gi or (ei and a and b');
eo <= ei and ((a AND b) OR (a' AND b'))
end compdf;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Code for the 4 bit comparator
entity mag_comp_4bit is
port(x, y : in std_logic_vector(3 downto 0);
ig, ie : in std_logic ;
og, oe : out std_logic);
end mag_comp_4bit;
architecture Comp_Struc of mag_comp_4bit is
component mag_comp_1bit is
end component ;
architecture Comp_Struc of mag_comp_4bit is
component mag_comp_1bit is
end component ;
architecture Comp_Struc of mag_comp_4bit is
component mag_comp_1bit is
end component ;
17.21 (a) \[ \begin{align*} 
& a' = \text{si AND a} \\
& b = \text{si AND a}
\end{align*} \]

17.21(c)

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity pr_sel_1bit is
  port (a, si : in std_logic;
        b, so : out std_logic);
end pr_sel_1bit;

architecture prdf of pr_sel_1bit is
begin
  so <= si and not a;
  b <= si and a;
end prdf;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Code for the 4 bit priority selector

entity pr_sel_4bit is
  port (x : in std_logic_vector(3 downto 0);
        isel : in std_logic;
        y : out std_logic_vector(3 downto 0);
        oisel : out std_logic);
end pr_sel_4bit;

architecture Pr_Struc of pr_sel_4bit is
  component pr_sel_1bit
    port (a, si : in std_logic;
          b, so : out std_logic);
  end component;

  signal sel : std_logic_vector(3 downto 0);

  pr_sel_1bit_3 : pr_sel_1bit
  port map (x(3), sel(3), y(3), sel(2));
  pr_sel_1bit_2 : pr_sel_1bit
  port map (x(2), sel(2), y(2), sel(1));
  pr_sel_1bit_1 : pr_sel_1bit
  port map (x(1), sel(1), y(1), sel(0));
  pr_sel_1bit_0 : pr_sel_1bit
  port map (x(0), sel(0), y(0), oisel);

  sel(3) <= isel;
end Pr_Struc;
```

17.21(b)

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

datatype myrom
constant rom8_3 :=
  letter '001' to '110';

entity pr_sel_1bit is
  port (a, si : in std_logic;
        b, so : out std_logic);
end pr_sel_1bit;

architecture prdf of pr_sel_1bit is
begin
  so <= si and not a;
  b <= si and a;
end prdf;
```

17.21 (d) Time X isel Y oisel sel
0 ns 1000 '1' 1000 '0' 1000
5 ns 0111 '1' 0100 '0' 1100
10 ns 0000 '1' 0000 '1' 1111

17.22

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity sm1 is
  port (x, clk : in std_logic;
        z : out std_logic);
end sm1;

architecture Behavioral of sm1 is
  type rom8_3 is array(0 to 7) of std_logic_vector(3 downto 0);
  constant myrom : rom8_3 :=
    letter "001" to "110";

  signal index, romout: std_logic_vector(0 to 2);
  signal q, d: std_logic_vector(1 to 2) := "00";

  begin
    index <= x & q;
    romout <= myrom(conv_integer(index));
    z <= romout(0);
    d <= romout(1 to 2);

    process(clk)
    begin
      if clk'event and clk='1' then q <= d; end if;
    end process;

end Behavioral;
```

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
The state assignment is as follows:
- \( q_0 q_1 q_2 q_3 \) -
- State 0 - 1000; State 1 - 0100; State 2 - 0010; State 3 - 0001

The VHDL code using equations derived by inspection from the state graph:

```vhdl
entity sm1 is
    port (x, clk : in bit;
          z : out bit);
end sm1;
architecture equations of sm1 is
    signal q0 : bit := '1';
    signal q1, q2, q3 : bit := '0';
    begin
        process(clk)
            begin
                if clk'event and clk='1' then
                    q0 <= (x and q0) or (not x and q1) or (not x and q3);
                    q1 <= (not x and q0) or (x and q3);
                    q2 <= (x and q2) or (x and q1);
                    q3 <= not x and q2;
                end if;
        end process;
        z <= (not x and q1) or (x and q3) or q2;
    end equations;
```

There are three problems with this code:
1) The sensitivity list for the process contains the signal select. It should be sel. (Select is a VHDL reserved word).
2) When sel is true, there are two assignments to muxsel in the process and only the second one has any effect. Hence, if sel is true for two successive executions of the process, muxsel will be incremented to 2.
3) Since muxsel is not changed until the process terminates, the selection uses the old value of muxsel not the new value.

### Table 17.25

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>Output</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>( S_0 )</td>
<td>10</td>
<td>( S_0 )</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>( S_1 )</td>
<td>01</td>
<td>( S_1 )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( S_2 )</td>
<td>01</td>
<td>( S_2 )</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>( S_0 )</td>
<td>00</td>
<td>( S_3 )</td>
</tr>
</tbody>
</table>

### Table 17.26

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>( S_0 )</td>
<td>1</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>( S_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( S_0 )</td>
<td>0</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>( S_0 )</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 17.27

<table>
<thead>
<tr>
<th>State</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>( S_0 )</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>( S_1 )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( S_2 )</td>
</tr>
</tbody>
</table>
17.28 (a) Present State | xin = 0 | xin = 1 | zout
--- | --- | --- | ---
S_1 | S_2 | S_10 | 0
S_2 | S_2 | S_3 | 0
S_3 | S_2 | S_3 | 1
S_4 | S_2 | S_3 | 0
S_5 | S_3 | S_10 | 1
S_6 | S_3 | S_10 | 0
S_7 | S_2 | S_10 | 0
S_8 | S_2 | S_10 | 0
S_9 | S_2 | S_10 | 0
S_10 | S_2 | S_10 | 0

17.28 (c) Present State | xin = 0 | xin = 1 | zout
--- | --- | --- | ---
S_1 | S_2 | S_3 | 0
S_2 | S_2 | S_3 | 0
S_3 | S_2 | S_3 | 1
S_4 | S_2 | S_3 | 0
S_5 | S_3 | S_10 | 0
S_6 | S_3 | S_10 | 0
S_7 | S_2 | S_10 | 0
S_8 | S_2 | S_10 | 0
S_9 | S_2 | S_10 | 0
S_10 | S_2 | S_10 | 0

17.28 (b) Signal | Current | 0ns | 40ns | 80ns | 120ns | 160ns | 200ns | 240ns | 280ns | 320ns | 360ns
--- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | ---
clk | '0' | | | | | | | | | | |
reset | '0' | | | | | | | | | | |
xin | '1' | | | | | | | | | | |
zout | '0' | | | | | | | | | | |
state | s10 | s1 | s2 | s3 | s4 | s8 | s5 | s10 | s9 | s8 | s5 | s1
nextstate | s10 | s2 | s3 | s6 | s4 | s7 | s8 | s5 | s10 | s9 | s8 | s5 | s10

17.28 (d) The output is 1 for an input sequence ending in either 01 or 1011

17.28 (e) Present State | xin = 0 | xin = 1 | zout
--- | --- | --- | ---
S_1 | S_2 | S_3 | 0
S_2 | S_2 | S_3 | 0
S_3 | S_2 | S_3 | 1
S_4 | S_2 | S_3 | 0
S_5 | S_3 | S_10 | 0
S_6 | S_3 | S_10 | 0
S_7 | S_2 | S_10 | 0
S_8 | S_2 | S_10 | 0
S_9 | S_2 | S_10 | 0
S_10 | S_2 | S_10 | 0

17.29 df1, df2, and df3 are the same. They have two flip-flops, y0 and y1; y1 has input xin and y0 has input the complement of y0. The output z is y0 AND (xin XOR y1) or equivalent AND-OR logic.
df4 is the same except that z is "registered" in a flip-flop.

17.29 (b) The output for df4 only changes on positive clock edges and is delayed with respect to the output for df1, df2 and df3. See the simulation waveforms below.

Simulation for df1, df2 and df3.

Simulation for df4.
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity mask_8 is
  port (X : in std_logic_vector(7 downto 0);
      Store, Set, Clk : in std_logic;
      Z : out std_logic_vector(7 downto 0));
end mask_8;
architecture Behavioral of mask_8 is
signal M : std_logic_vector(7 downto 0);
begin
  process(Set, Clk)
  begin
    if Set='1' then
      M <= "11111111";
    elsif Clk'event and Clk='1' then
      if Store='1' then M<=X; end if;
    end if;
  end process;
  Z <= M and X;
end Behavioral;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Seq_143 is
  port (Clk, X : in std_logic;
        Z : out std_logic);
end Seq_143;
architecture Moore of Seq_143 is
signal State : integer := 0;
signal NextState : integer range 0 to 3;
begin
  process(State, X)
  begin
    case State is
      when 0 => Z <= '0';
      if X = '0' then NextState <= 0; end if;
      else NextState <= 1; end if;
      when 1 => Z <= '0';
      if X = '0' then NextState <= 2; end if;
      else NextState <= 1; end if;
      when 2 => Z <= '0';
      if X = '0' then NextState <= 0; end if;
      else NextState <= 3; end if;
      when 3 => Z <= '1';
      if X = '0' then NextState <= 2; end if;
      else NextState <= 1; end if;
    end case;
  end process;
  process(Clk)
  begin
    if Clk'event and Clk='1' then
      State <= NextState; end if;
  end process;
end Moore;
Unit 18 Problem Solutions

18.3 See FLD p. 736 for circuit. Notice that the $Q$ output of the flip-flop is $b_{in}$, while the $D$ input is $b_{out}$.

18.4 See FLD p. 737. AND-ing with $x_i$ is like $M/Ad$ if $x_i$ is 1. Shifting is like moving from AND gates involving $x_i$ to those involving $x_{i-1}$ or from $x_i$ to $x_{i+1}$.

18.5 See FLD p. 737. Compare to divider state graph of FLD Figure 18-11.

18.6 See FLD p. 737.

18.7 (a) Overflow occurs only on division by 0, so $V = y_0'y_1'y_2'y_3' = (y_0 + y_1 + y_2 + y_3)'$

18.7 (b) See FLD p. 738.

18.8 See FLD p. 738.

Notes: 1) The value in the carry FF does not matter for the first addition. 2) Only a half-adder is needed since all the additions are of two bits.

Next State Table: $S_0 = 00, S_1 = 01, S_2 = 11$

<table>
<thead>
<tr>
<th>$Q_1Q_0$</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>00</td>
<td>00</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

$D_1 = (Q_1' + Z')Q_0$

$D_0 = (S_1 + Q_0')(Q_1' + Z')$

$Q_1'Q_0 + Z'Q_0 + S_1Q_1'$

Output Table: $L$ Sh Dec C

<table>
<thead>
<tr>
<th>$Q_1Q_0$</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0000</td>
<td>0000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>01</td>
<td>0111</td>
<td>0111</td>
<td>0111</td>
<td>0111</td>
</tr>
<tr>
<td>11</td>
<td>0110</td>
<td>0000</td>
<td>0000</td>
<td>0110</td>
</tr>
<tr>
<td>10</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
</tbody>
</table>

$L = (S_1' + Q_0')$

$Sh = Dec = (Q_1' + Z')Q_0$

$C = (Q_1 + Q_0)'$
Unit 18 Solutions

18.10

Notes: 1) The carry FF must contain 0 for the first addition. 2) Only a half-adder is needed since all the additions are of two bits; the first addition will produce a 0 carry.

Next State Table: \( S_0 = 00, S_1 = 01, S_2 = 11, S_3 = 10 \)

\[
\begin{array}{c|cccc|c|cccc}
Q_1Q_0 & 00 & 01 & 11 & 10 & St & Z \\
\hline
00 & 00 & 00 & 01 & 01 & 00 & 00 & 00 & 01 & 01 \\
11 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 \\
\end{array}
\]

\[
D_1 = (Q_0 + Z')(Q_0 + Q_1) \\
D_0 = (Q_0 + St)Q_1'
\]

Output Table: L Sh Dec C

\[
\begin{array}{c|cccc|c|cccc}
Q_1Q_0 & 00 & 01 & 11 & 10 & St & Z \\
\hline
00 & 0000 & 0000 & 1000 & 1000 & 0000 & 0000 & 0000 & 0110 & 0110 \\
01 & 0110 & 0110 & 0110 & 0110 & 0110 & 0110 & 0110 & 0110 & 0110 \\
11 & 0111 & 0111 & 0111 & 0111 & 0111 & 0111 & 0111 & 0111 & 0111 \\
10 & 0110 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 & 0110 \\
\end{array}
\]

\[
L = (St' + Q_1 + Q_0)' \\
Sh = Dec = (Q_0 + Z')(Q_0 + Q_1) \\
C = (Q_1' + Q_0)' 
\]

18.11

Notes: 1) The carry FF must contain 0 for the first addition. 2) A full-adder is needed since the second addition may be of three bits. 3)The next state and output equations are the same as in 18.10 except \( C = Q_0 \).
18.12 (a) Inputs and outputs are given in decimal in the table. Inputs 10 through 14 are assumed to never occur.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A, 0</td>
<td>B, 9</td>
<td>B, 8</td>
<td>B, 7</td>
<td>B, 6</td>
<td>B, 5</td>
<td>B, 4</td>
<td>B, 3</td>
<td>B, 2</td>
<td>B, 1</td>
<td>A, 15</td>
</tr>
<tr>
<td>B</td>
<td>B, 9</td>
<td>B, 8</td>
<td>B, 7</td>
<td>B, 6</td>
<td>B, 5</td>
<td>B, 4</td>
<td>B, 3</td>
<td>B, 2</td>
<td>B, 1</td>
<td>B, 0</td>
<td>A, 15</td>
</tr>
</tbody>
</table>

18.12 (b) Use the state assignment $Q = 0$ for state A and $Q = 1$ for state B. There are 159 minimum sum-of-product equations for $Q^+$; one solution is

$$Q^+ = X_3X_0 + X_3X_2 + X_3X_0' + X_3X_2' + QX_2'.$$

The minimum sum-of-product equations for the outputs are

$$Z_3 = X_3X_2'X_0 + QX_2X_2X_1 + X_3X_2'X_1' + QX_2X_0' + X_3X_0 + X_3X_2'X_0' + QX_2X_0' + QX_2X_1' + X_3X_1$$

18.13 (a) Inputs and outputs are given in decimal in the table. Inputs 1, 2, 13, 14, and 15 are assumed to never occur.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A, 0</td>
<td>A, 3</td>
<td>B, 12</td>
<td>B, 11</td>
<td>B, 10</td>
<td>B, 9</td>
<td>B, 8</td>
<td>B, 7</td>
<td>B, 6</td>
<td>B, 5</td>
<td>B, 4</td>
</tr>
<tr>
<td>B</td>
<td>B, 9</td>
<td>B, 12</td>
<td>B, 11</td>
<td>B, 10</td>
<td>B, 9</td>
<td>B, 8</td>
<td>B, 7</td>
<td>B, 6</td>
<td>B, 5</td>
<td>B, 4</td>
<td>B, 3</td>
</tr>
</tbody>
</table>

18.13 (b) Use the state assignment $Q = 0$ for state A and $Q = 1$ for state B. The minimum sum-of-product equations for $Q^+$ are

$$Q^+ = X_3 + X_2 + QX_0$$

18.14 The ONE ADDER is similar to a serial adder, except that there is only one input. This means that the carry will be added to $X$. Thus, if the carry flip-flop is initially set to 1, 1 will be added to the input. The signal $I$ can be used to preset the carry flip-flop to 1.

Let $S_0$ represent Carry = 0, and let $S_1$ represent Carry = 1. The state graph is as follows:

![State Graph](https://via.placeholder.com/150)

$$Q^+ = Q(Sh' + X)$$

$$Z = (Q + X)(Q' + X')(X + Sh)$$

$$Z = (Q + X)(Q' + X')(Q' + Sh)$$
Unit 18 Solutions

18.14 (contd)

18.15 (a)

18.15 (b)

18.15 (c)

18.15 (d)

18.15 (e)
18.15 (d) Present State | Next State | Ad Sh Load Done
<table>
<thead>
<tr>
<th>StM: 00 01 10 11</th>
<th>00 01 10 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$ $S_0$ $S_1$ $S_1$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_1$ $S_2$ $S_1$ $S_2$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_2$ $S_2$ $S_2$ $S_2$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_3$ $S_3$ $S_3$ $S_3$</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_4$ $S_4$ $S_4$ $S_4$</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_5$ $S_5$ $S_5$ $S_5$</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$S_6$ $S_6$ $S_6$ $S_6$</td>
</tr>
<tr>
<td>$S_7$</td>
<td>$S_7$ $S_7$ $S_7$ $S_7$</td>
</tr>
</tbody>
</table>

For this assignment, from LogicAid:

- $J_A = StB'C' + MC$; $K_A = M' + B + C$; $J_b = A'C$; $K_b = A'C'$; $J_c = AB'$; $K_c = A'B$;
- $Ad = MAB'C' + MA'C$; $Sh = M'A + M'C + AB + AC$; $Load = StA'B'C'$; $Done = A'BC'$

**Unit 18 Solutions**

I. $(S_0', S_1') (S_0, S_2) (S_3, S_5) (S_6, S_8)$

II. $(S_0', S_1') (S_0, S_2) (S_3, S_5) (S_6, S_8)$

III. $(S_1, S_3') (S_2, S_4) (S_6, S_8)$

(Other assignments are possible.)

For this assignment, from LogicAid:

- $J_A = StB'C' + MC$; $K_A = M' + B + C$; $J_b = A'C$; $K_b = A'C'$; $J_c = AB'$; $K_c = A'B$;
- $Ad = MAB'C' + MA'C$; $Sh = M'A + M'C + AB + AC$; $Load = StA'B'C'$; $Done = A'BC'$

**Diagram:**

- 4-bit multiplier
- 3-bit adder
- 7 flip-flops (Q0 to Q6)
- Clock (Clk)
- Load (Ld)
- Ad
- Sh
- St
- M

OR gates, AND gates, & inverters implement the equations from 18.15 (d)
Unit 18 Solutions

18.16 (a)

\[ \text{product} \]

\[ \begin{array}{cccccccc}
8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
\hline
\text{ACC} & \text{multiplier} & \text{5-BIT ADDER} & \text{multiplicand} & \text{Load} & \text{Sh} & \text{Ad} & \text{Clk} & \text{C5} \\
\end{array} \]

18.16 (b) See solution to 18.15 (b).

18.16 (d) Graph is same as 18.15, so from LogicAid, using the same state assignment:

\[
D_A = \text{St}A'B'C' + MAB'C' + MA'C
\]
\[
D_B = A'C + AB
\]
\[
D_C = AB' + B'C + AC
\]

Ad, Sh, Ld, Done: See solution to 18.15 (d)

18.16 (c)

<table>
<thead>
<tr>
<th>St M A B C</th>
<th>D_A D_B D_C Ad Sh Ld Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 0 0 0 0 1 0</td>
<td>1 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>- 1 1 0 0 0 1 0 0 0</td>
<td>1 0 0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>- 1 0 - 1 1 0 0 1 0 0 0</td>
<td>1 0 1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>- - 0 - 1 0 1 0 0 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>- - 1 1 - 0 1 0 0 1 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>- - 1 0 - 0 1 0 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>- - 0 1 - 0 0 1 0 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>- - 1 1 0 0 0 1 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>- - 0 - 1 1 0 0 0 0 1 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>- 1 0 1 0 0 0 0 1 0 0 0</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>- 1 0 0 0 0 0 0 0 0 0 0 1</td>
<td>1 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

18.16 (c) 0 0 0 0 0 0 1 1 0 shift
0 0 0 0 0 0 1 1 1 shift
1 0 1 0 0 add
0 0 1 0 1 0 0 0 1 add
0 1 0 1 0 0 shift
0 1 1 1 1 0 0 0 1 shift
0 0 1 1 1 1 0 0 0 shift
18.16 (d) (contd)

18.17 (a)

18.17 (b)

18.18 (a) (alternate solution)
Unit 18 Solutions

18.18 (b)  
\[
\begin{array}{cccccccc}
  x_7 & x_6 & x_5 & x_4 & x_3 & x_2 & x_1 & x_0 \\
  \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
  \text{Subtractor-Comparator} & C & \text{Shift} & \text{Load} & \text{Control} & \text{St} & \text{St'} & \text{Clk} \\
  \end{array}
\]

18.18 (d)  
\[
\begin{array}{cccccccc}
  0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
  1 & 1 \\
  \text{shift } C = 0 \\
  \hline
  1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
  1 & 1 \\
  \text{sub. } C = 1 \\
  \hline
  0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
  1 & 1 \\
  \text{shift } C = 0 \\
  \hline
  0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
  1 & 1 \\
  \text{sub. } C = 1 \\
  \hline
  0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
  1 & 1 \\
  \text{shift } C = 0 \\
  \hline
  1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
  1 & 1 \\
  \text{sub. } C = 1 \\
  \hline
  0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
  1 & 1 \\
  \text{shift } C = 0 \\
  \hline
  0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
  \text{remainder quotient}
\end{array}
\]

18.18 (e)  
\[
\begin{array}{cccccccc}
  \text{S0} & \text{S1} & \text{S2} & \text{S3} & \text{S4} & \text{S5} & \text{S6} \\
  \text{C} & \text{C} & \text{C} & \text{C} & \text{C} & \text{C} & \text{C} \\
  \text{Su} & \text{Su} & \text{Su} & \text{Su} & \text{Su} & \text{Su} & \text{Su} \\
  \text{Sh} & \text{Sh} & \text{Sh} & \text{Sh} & \text{Sh} & \text{Sh} & \text{Sh} \\
  \text{C} & \text{C} & \text{C} & \text{C} & \text{C} & \text{C} & \text{C} \\
  \text{V} & \text{V} & \text{V} & \text{V} & \text{V} & \text{V} & \text{V} \\
  \text{St} & \text{St} & \text{St} & \text{St} & \text{St} & \text{St} & \text{St} \\
  \text{Ld} & \text{Ld} & \text{Ld} & \text{Ld} & \text{Ld} & \text{Ld} & \text{Ld} \\
\end{array}
\]

18.19 (a)  
\[
\begin{array}{cccccccc}
  x_7 & x_6 & x_5 & x_4 & x_3 & x_2 & x_1 \\
  \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
  \text{Subtractor-Comparator} & C & \text{Shift} & \text{Load} & \text{Control} & \text{St} & \text{St'} & \text{Clk} \\
  \end{array}
\]

18.19 (b)  
\[
\begin{array}{cccccccc}
  \text{S0} & \text{S1} & \text{S2} & \text{S3} & \text{S4} & \text{S5} & \text{S6} \\
  \text{C} & \text{C} & \text{C} & \text{C} & \text{C} & \text{C} & \text{C} \\
  \text{Su} & \text{Su} & \text{Su} & \text{Su} & \text{Su} & \text{Su} & \text{Su} \\
  \text{Sh} & \text{Sh} & \text{Sh} & \text{Sh} & \text{Sh} & \text{Sh} & \text{Sh} \\
  \text{C} & \text{C} & \text{C} & \text{C} & \text{C} & \text{C} & \text{C} \\
  \text{V} & \text{V} & \text{V} & \text{V} & \text{V} & \text{V} & \text{V} \\
  \text{St} & \text{St} & \text{St} & \text{St} & \text{St} & \text{St} & \text{St} \\
  \text{Ld} & \text{Ld} & \text{Ld} & \text{Ld} & \text{Ld} & \text{Ld} & \text{Ld} \\
\end{array}
\]

18.19 (c)  
\[
\begin{array}{cccccccc}
  \text{d}_7 & \text{d}_6 & \text{d}_5 & \text{d}_4 & \text{d}_3 \\
  \text{F.A.} & \text{F.A.} & \text{F.A.} & \text{F.A.} & \text{F.A.} \\
  \text{Comparator C} & \text{Comparator C} & \text{Comparator C} & \text{Comparator C} & \text{Comparator C} \\
  \end{array}
\]

© 2010 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.
18.19 (d)  

\[
\begin{array}{ccccccc}
0 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & & & \\
1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & & & \\
1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\text{shift } C = 0 \\
\text{sub. } C = 1 \\
\text{shift } C = 0 \\
\text{sub. } C = 1 \\

\text{remainder} \quad \text{quotient}

18.20 (b)  
\[D_0 = St'Q_1 + KQ_1 + KQ_2; \quad D_1 = StQ_0 + K'B'Q_1 + K'BQ_2; \quad D_2 = K'BQ_1 + K'B'Q_2; \quad R = StQ_0 \]
\[Sh = K'B'Q_1 + K'BQ_2 + K'B'Q_2 = K'Q_1 + K'Q_2; \quad X = KQ_1 + K'BQ_1 + K'BQ_2 = KQ_1 + BQ_1 + K'BQ_2\]

18.21 (a)  

\[\text{Counter} \quad \text{Controller} \quad \text{Clk} \quad \text{St} \quad \text{Clk} \quad \text{Er} \quad \text{SI} \quad \text{Sh} \quad \text{K}\]

18.21 (b)  

\[\text{Shift Register A} \quad \text{Shift Register B} \quad \text{Logic Circuit} \quad \text{C} \quad \text{SI} \quad \text{a} \quad \text{b} \quad \text{Clk} \quad \text{St} \quad \text{Sh} \quad \text{Clk} \quad \text{K}\]

18.21 (c)  
\[D_1 = St'Q_1 + KQ_2 + KQ_3 \quad \text{Clr}= StQ_1 \]
\[D_2 = K'S'_0Q_2 + StQ_1 \quad \text{Sh} = K'S'_0Q_2 + K'S'_0Q_2 + K'SQ_2 + K'Q_3 \]
\[D_3 = K'S'_0Q_2 + K'Q_3 \quad \text{Er}= KSQ_2 \]
\[\text{SI} = K'S'_0Q_2 + K'S'_0Q_3\]

Note: The signal marked by ↓ is the shift register serial output, not a state.

18.22 (a)
Unit 18 Solutions

18.22 (b)

18.22 (c) I. \((S_0, S_1)\times2\) \((S_1, S_2)\) \((S_0, S_2)\)
II. \((S_0, S_1)\times2\) \((S_1, S_0)\) \((S_0, S_0)\times2\)
From Karnaugh maps:
\[ D_0 = Q_0^* = StQ_0 + KQ_0Q_1 \]
\[ D_1 = Q_1^* = St; \quad Sh = StQ_0 \]
Alternative: \(Q_0^* = StQ_0 + StKQ_1\)

<table>
<thead>
<tr>
<th>(St)</th>
<th>(K)</th>
<th>(Q_0)</th>
<th>(Q_1)</th>
<th>(D_0)</th>
<th>(D_1)</th>
<th>(Sh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 1 0 1 1 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 0 0 0 0 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

18.22 (d) \(SI = C'ab + Cab' + Ca'b\)

18.23 (a)

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>Reset</td>
</tr>
<tr>
<td>(S_1)</td>
<td>Find AND of (A &amp; B)</td>
</tr>
<tr>
<td>(S_2)</td>
<td>Find XOR of (A &amp; B)</td>
</tr>
</tbody>
</table>
18.23 (c) \( Q_0^* = St'CQ_0 + K'Q_1 + KQ_2; \) \( Q_1^* = StCQ_0 + K'Q_1; \) \( Q_2^* = St'CQ_0 + K'Q_2; \)

\( Sh = StCQ_0 + St'CQ_2 + K'Q_1 + K'Q_2 + KQ_2 \)

\( D = StCQ_0 + K'Q_1 + KQ_1 \)

<table>
<thead>
<tr>
<th>St</th>
<th>C</th>
<th>K</th>
<th>Q_0</th>
<th>Q_1</th>
<th>Q_2</th>
<th>Q_0^*</th>
<th>Q_1^*</th>
<th>Q_2^*</th>
<th>Sh</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

18.23 (d) Change \( C' \) to \( D' \) in 18.22 (d)

\( SI = D'ab + Dab' + Da'b \)

18.24 (a)

18.24 (b)

Note: \( M \) can be determined independently of the state of the system, so it is not included in the state graph.

18.25 (a)

18.25 (b)

18.25 (c)

<table>
<thead>
<tr>
<th>State</th>
<th>St ( \times )</th>
<th>ABl</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00 01 11 10</td>
<td>00 01 11 10</td>
<td></td>
</tr>
<tr>
<td>( S_0 )</td>
<td>( S_0 )</td>
<td>( S_0 )</td>
<td>( S_0 )</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>( S_1 )</td>
<td>( S_1 )</td>
<td>( S_1 )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( S_2 )</td>
<td>( S_2 )</td>
<td>( S_2 )</td>
</tr>
</tbody>
</table>

\( D_1 = KQ_3 + K'Q_i; \) \( D_2 = St + K'Q_2; \) \( A = K'Q_1; \)

\( B = K'Q_2; \) \( Ld = St + KQ_2 \)
Unit 18 Solutions

18.26

18.27 (a)

18.27 (b) \( J = ST; \ K = ZER1 \ ZER2; \)

\[
\begin{align*}
\text{Done} &= ZER1 \ ZER2 \ Q; \\
\text{CLR} &= STQ'; \\
LD2 &= STQ'; \\
LD1 &= STQ' + ZER1 \ ZER2' \ Q; \\
CT1 &= ZER1' \ Q; \\
CT2 &= ZER1 \ ZER2' \ Q.
\end{align*}
\]

18.27 (c) \((N_1 + 1)N_2 \text{ cycles}\)

18.28 (a) Initial PU, PL: 0000 0000

1st Add Lower half PU, PL: 0000 1011
1st Add Upper half PU, PL: 0000 1011
2nd Add Lower half PU, PL: 0000 0110
2nd Add Upper half PU, PL: 0001 0110
3rd Add Lower half PU, PL: 0001 0001
3rd Add Upper half PU, PL: 0010 0001
4th Add Lower half PU, PL: 0010 1100
4th Add Upper half PU, PL: 0010 1100
5th Add Lower half PU, PL: 0010 0111
5th Add Upper half PU, PL: 0011 0111

18.28 (c) Label the 4 FF outputs \( S_0, S_1, S_2 \) and \( S_3 \).

\[
\begin{align*}
D_0 &= S'S_0 + S'S_3 \\
D_1 &= S(S_0) + S_2 \\
D_2 &= (BZ)S_1 \\
D_3 &= (BZ)S_1 + S(S_1) \\
CP &= LA = LB = CC = S_0 \\
LPL &= EA = DB = (BZ)S_1 \\
LPU &= MS = S_2 \\
D &= S_3
\end{align*}
\]

18.28 (b) Assume two FFs \( Q_1Q_0 \) and the following encoding:

\[
\begin{align*}
S_0 &= 00, \ S_1 = 01, \ S_2 = 11, \text{ and } S_3 = 10. \text{ Then,} \\
D_0 &= S(S_0) + S_2 + (BZ)S_1 \\
&= SQ_1Q_0' + Q_1Q_0 + (BZ)Q_1'Q_0 \\
D_1 &= (BZ)S_1 + (BZ)S_1 + S(S_1) \\
&= S_1 + S(S_2) \\
&= Q_1Q_0' + SQ_0Q_0' \\
CP &= LA = LB = CC = S_0 = Q_1Q_0' \\
LPL &= EA = DB = (BZ)S_1 = (BZ)Q_1'Q_0 \\
LPU &= MS = S_2 = Q_1Q_0 \\
D &= S_3 = Q_1Q_0' 
\end{align*}
\]
18.29 (a) When the multiplier is negative, the B counter can be incremented to zero. Two control inputs are assumed: DB (decrement B) and IB (Increment B). Also, when the multiplier is negative, the multiplicand must be subtracted to produce the product. This can be done by adding an Exclusive-OR array after the AND array; when IA = 0, the output of the Exclusive-OR array is equal to its input and when IA = 1, it inverts its input. To produce a two's complement subtract, the carry FF must be set; SC (set carry) has been added to the carry FF. When the product is negative, \((A_3 \oplus B_3) = 1\), IA must be 1 to extend the sign bit when adding the carry to the upper half of the product.

18.29 (b) Answer is the same for both parts of Part (b).

\[
\begin{align*}
\text{Initial PU, PL:} & \quad 0000 \ 0000 \\
1\text{st Add Lower half PU, PL:} & \quad 0000 \ 1011 \\
2\text{nd Add Lower half PU, PL:} & \quad 1111 \ 0111 \\
3\text{rd Add Lower half PU, PL:} & \quad 1111 \ 0001 \\
4\text{th Add Lower half PU, PL:} & \quad 1111 \ 1100 \\
5\text{th Add Lower half PU, PL:} & \quad 1110 \ 1100 \\
1\text{st Add Upper half PU, PL:} & \quad 1111 \ 1011 \\
2\text{nd Add Upper half PU, PL:} & \quad 1111 \ 0001 \\
3\text{rd Add Upper half PU, PL:} & \quad 1111 \ 1100 \\
4\text{th Add Upper half PU, PL:} & \quad 1110 \ 1100 \\
5\text{th Add Upper half PU, PL:} & \quad 1110 \ 0111 \\
\end{align*}
\]

18.29 (c) Label the 5 FF outputs \(S_0, S_1, S_2, S_3, S_4\).

\[
\begin{align*}
D_0 &= S'S_0 + SS_4, \quad D_1 = S(S_0), \quad D_2 = S_1 + S_3 \\
D_3 &= (BZ)S_2, \quad D_4 = (BZ)S_2 + S(S_4) \\
CP &= LA = L = S_0 \\
CC &= B_3(S_1 + S_3), \quad SC = B_3(S_1 + S_3) \\
LPL &= EA = (BZ)S_2 \\
DB &= B_3'S_3, \quad IB = B_3S_3 \\
IA &= B_3(BZ)'S_2 + (A_3 \oplus B_3)S_3 \\
LPU &= MS = S_3 \\
D &= S_4
\end{align*}
\]

Note: The PU and PL registers are connected the same way as in Problem 18.28.

18.29 (d)
18.30 (a) $D = \text{EZERO}' Q + \text{St}' Q'$; Done = EZERO Q; CLR = St' Q'; LOAD = St Q' + \text{IZERO EZERO}' Q$

18.30 (b) $D = \text{EZERO}' Q + \text{St}' Q'$; Done = EZERO Q; CLR = St' Q'; LOAD = St Q' + \text{IZERO EZERO}' Q$

18.30 (c) $N_1 + (N_1/N_2)$ cycles (round down)

18.30 (d) The quotient counter reaches 1111, and $UP = 1$ again.

18.30 (e) The quotient will count upward forever, and Done will never be 1.

Note: These equations simplify because 101, 110 and 111 are don't-care state combinations.

---

Unit 18 Solutions

18.29 (e) Assume three FFs $Q_2Q_1Q_0$ and the following encoding: $S_0 = 000, S_1 = 001, S_2 = 011, S_3 = 010$ and $S_4 = 100$. Then,

$$
D_0 = S(S_0) + S_1 + S_3 = SQ_2'Q_0' + Q_2Q_1'Q_0 + S_3Q_2'Q_0' = SQ_2'Q_0' + Q_1'Q_0 + Q_1Q_0' \\
D_1 = S_1 + S_3 + (BZ'S_2 = Q_2Q_1'Q_0 + Q_2Q_1Q_0' + (BZ)'Q_2Q_1Q_0 = Q_1'Q_0 + Q_1Q_0' + (BZ)'Q_1 \\
D_2 = (BZ)S_2 + S(S_0) = (BZ)Q_2Q_1Q_0 + S(Q_2Q_1Q_0) = (BZ)Q_1Q_0 + S(Q_2)
$$

18.30 (d) The quotient counter reaches 1111, and $UP = 1$ again.

18.30 (e) The quotient will count upward forever, and Done will never be 1.

18.31

When the done signal comes on, square root is in the 4-bit counter.
Unit 19 Problem Solutions

19.1  See FLD p. 739 for solution.

19.3  See FLD p. 739 for solution.

19.5  See FLD p. 740 for solution.

19.7  See FLD p. 741 for solution.

19.9  See FLD p. 741-742 for solution.

19.11

19.12 (a)

19.12 (b)

19.2  See FLD p. 739 for solution.

19.4  See FLD p. 740 for solution.

19.6  See FLD p. 741 for solution.

19.8  See FLD p. 741 for solution.

19.10 See FLD p. 742 for solution.
19.14 (b) Let $S_0$, $S_1$, $S_2$ and $S_3$ be the four FF outputs, then
$D_0 = x(S_0 + S_1 + S_2 + S_3)$,
$D_1 = xS_0$,
$D_2 = xS_1$, and
$D_3 = x(S_2 + S_3)$.
$Z = S_3$

19.14 (d) Using the simplification identity twice,
$D_1 = x(Q_0 + Q_1)$ and $D_0 = x(Q_0' + Q_1)$.

19.14 (c) Using state assignment $S_0 = 00$, $S_1 = 01$, $S_2 = 10$ and $S_3 = 11$, and denoting state variables $Q_1$ and $Q_0$, $D_1$ is the OR of $D_2$ and $D_3$ from Part b) so
$D_1 = x(S_1 + S_2 + S_3)$
$= x(Q_1 Q_0' + Q_1 Q_0 + Q_1 Q_0')$.
Similarly, $D_0$ is the OR of $D_1$ and $D_3$ from Part b) so
$D_0 = x(S_0 + S_2 + S_3)$
$= x(Q_1 Q_0' + Q_1 Q_0' + Q_1 Q_0)$.
$Z = Q_1 Q_0$
19.15 (a) Reset

19.15 (b) Next State Table for $Q_1^+ Q_0^+$: $S_1 = 00$, $S_2 = 01$ and $S_3 = 11$.

<table>
<thead>
<tr>
<th>$Q_1 Q_0$</th>
<th>000</th>
<th>001</th>
<th>011</th>
<th>010</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>11</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>10</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Q_1 Q_0$</th>
<th>100</th>
<th>101</th>
<th>111</th>
<th>110</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>11</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>10</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

$D_1 = K'Q_1 + KS_0Q_0$, $D_0 = K'Q_0 + StQ_0'$

Output Table for $Clr$ $Sh$ $Er$ $SI$

<table>
<thead>
<tr>
<th>$Q_1 Q_0$</th>
<th>000</th>
<th>001</th>
<th>011</th>
<th>010</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0000</td>
<td>0000</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>01</td>
<td>0100</td>
<td>0101</td>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>11</td>
<td>0101</td>
<td>0100</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>10</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Q_1 Q_0$</th>
<th>100</th>
<th>101</th>
<th>111</th>
<th>110</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>01</td>
<td>0100</td>
<td>0101</td>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>11</td>
<td>0101</td>
<td>0100</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>10</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

$Clr = StQ_0'$; $Sh = K'S_0Q_0 + Q_1 Q_0$

$Er = KS_0Q_0'$; $SI = KS_0Q_1 + KS_0Q_1 Q_0$

19.15 (c) Label the three FF outputs $S_1$, $S_2$ and $S_3$.

$D_1 = St'S_1 + KS_2 + KS_3$; $D_2 = K'S_0' S_2 + StS_1$

$D_3 = K'S_0 S_2 + K'S_3$; $Clr = StS_1$

$Sh = K'S_0' S_2 + K'S_2 + K'S_3 = K'S_2 + S_0' S_2 + K'S_3$

$Er = KS_0' S_2$; $SI = K'S_0 S_2 + K'S_0' S_3$

19.15 (d) Label the two FF outputs $Q_1$, $Q_0$ and the decoder outputs $S_1 = 00$, $S_2 = 01$ and $S_3 = 11^*$, then

$D_0 = K'S_0' S_2 + StS_1 + K'S_0 S_2 + K'S_3$

$= K'S_0 S_2 + StS_1 + K'S_3$

$D_1 = K'S_0 S_2 + K'S_3$; $Clr = StS_1$

$Sh = K'S_0' S_2 + K'S_2 + K'S_3$

$Er = KS_0' S_2$; $SI = K'S_0' S_2 + K'S_0' S_3$

*Note: Other solutions are possible for different encodings.
Unit 19 Solutions

19.16

19.17

19.18

19.19 (a)

19.19 (b)

\[ A^* = ABX + ABX'(X_1' + X_2') + \{AB\} \]
\[ = BX + AX'(X_1' + X_2') \]
\[ B^* = AB'(X_1' + X_2') + ABX_1' + A'BX_2' + \{AB\} \]
\[ = AX_1' + A'BX_1' + AX_2' \]
\[ Z_1 = A + B + X_2; \quad Z_2 = A'BX_1'; \quad Z_3 = ABX_2'X_1' \]

In the preceding equations, curly brackets (\{\}) indicate a don’t care term.
19.19 (c) PLA table obtained by tracing link paths:

<table>
<thead>
<tr>
<th>State</th>
<th>( A )</th>
<th>( B )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( X_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>00</td>
<td>00</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>01</td>
<td>00</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>10</td>
<td>00</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

19.19 (d) \( 2^3 \times 5 \) ROM

<table>
<thead>
<tr>
<th>( A )</th>
<th>( B )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( X_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>000</td>
<td>00</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>001</td>
<td>01</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>010</td>
<td>10</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>011</td>
<td>10</td>
<td>101</td>
<td></td>
</tr>
</tbody>
</table>

19.20 (a) In order to repeat the pattern 12 times, the counter must be loaded with 0011 in state \( S_0 \). Thus, \( LDN \) must be 0 in \( S_0 \) and 1 in \( S_1 \) and \( S_2 \). (It can be either 0 or 1 in \( S_3 \).)

19.20 (b) Let \( S_0, S_1, S_2 \) and \( S_3 \) be the four FF outputs, then

\[
D_0 = s'(S_0 + S_3); \quad D_1 = sS_0 + (TC)S_2 \\
D_2 = S_1; \quad D_3 = (TC)S_2 \\
LDN = S_1 + S_2 \text{ or } LDN = S_1 + S_2 + S_3 \\
CE = S_1 \text{ or } CE = S_1 + S_3 \\
z_1 = S_1; \quad z_2 = S_2 \\
P_3 = 0; \quad P_2 = 0; \quad P_1 = 1; \quad P_0 = 1
\]

Similarly, \( D_0 \) is the OR of \( D_1 \) and \( D_2 \) from Part (b) so

\[
D_0 = sS_0 + (TC)S_2 + S_1 \\
LDN = S_1 + S_2 \text{ or } LDN = S_1 + S_2 + S_3 \\
CE = S_1 \text{ or } CE = S_1 + S_3 \\
z_1 = S_1; \quad z_2 = S_2 \\
P_3 = 0; \quad P_2 = 0; \quad P_1 = 1; \quad P_0 = 1
\]

19.20 (c) Using state assignment \( S_0 = 00, S_1 = 01, S_2 = 11, S_3 = 10 \), and denoting the state variables as \( Q_1 Q_0 \), \( D_1 \) is the OR of \( D_2 \) and \( D_3 \) from Part (b) so

\[
D_1 = S_1 + (TC)S_2 = Q_1'Q_0 + (TC)Q_1Q_0 \\
D_0 = sS_0 + (TC)S_2 + S_1 \\
LDN = S_1 + S_2 \text{ or } LDN = S_1 + S_2 + S_3 \\
CE = S_1 \text{ or } CE = S_1 + S_3 \\
z_1 = S_1; \quad z_2 = S_2 \\
P_3 = 0; \quad P_2 = 0; \quad P_1 = 1; \quad P_0 = 1
\]

Similarly, \( D_0 \) is the OR of \( D_1 \) and \( D_2 \) from Part (b) so

\[
D_0 = sS_0 + (TC)S_2 + S_1 \\
LDN = S_1 + S_2 \text{ or } LDN = S_1 + S_2 + S_3 \\
CE = S_1 \text{ or } CE = S_1 + S_3 \\
z_1 = S_1; \quad z_2 = S_2 \\
P_3 = 0; \quad P_2 = 0; \quad P_1 = 1; \quad P_0 = 1
\]

The outputs are

\[
LDN = S_1 + S_2 = Q_0; \quad CE = S_1 = Q_1'Q_0 \\
z_1 = Q_1'Q_0 \text{ and } z_2 = Q_1Q_0
\]

19.21 (a) Initial PU,PL: 0000 0000

- 1st Add Lower half PU, PL: 0000 0110
- 1st Add Upper half PU, PL: 0000 1110
- 2nd Add Lower half PU, PL: 0000 0110
- 2nd Add Upper half PU, PL: 0001 0110
- 3rd Add Lower half PU, PL: 0000 0001
- 3rd Add Upper half PU, PL: 0010 0001
- 4th Add Lower half PU, PL: 0011 1000
- 4th Add Upper half PU, PL: 0010 1100
- 5th Add Lower half PU, PL: 0010 0111
- 5th Add Upper half PU, PL: 0011 0111
19.21 (b) Label the 4 FF outputs $S_0$, $S_1$, $S_2$ and $S_3$.

$$D_0 = S'S_0 + S'S_3$$
$$D_1 = S(S_0) + S_2$$
$$D_2 = (BZ)S_1$$
$$D_3 = (BZ)S_1 + S(S_3)$$
$$CP = LA = LB = CC = S_0$$
$$LPL = EA = DB = (BZ)'S_1$$
$$LPU = MS = S_2$$
$$D = S_3$$

19.21 (c) Assume two FFs $Q_1Q_0$ and the following encoding:

- $S_0 = 00$, $S_1 = 01$, $S_2 = 11$ and $S_3 = 10$. (The decoder outputs are labeled $S_0$, $S_1$, $S_2$ and $S_3$.)

Then,

$$D_0 = S(S_0) + S_2 + (BZ)'S_1$$
$$D_1 = (BZ)'S_1 + (BZ)S_1 + S(S_3) = S_1 + S(S_3)$$

The output equations are the same as in Part (c).

* Although SB is 1 in state $S_2$, shifting does not occur until the next clock.
19.23 (a) See answer to 18.30 (b).

19.24 (a) See answer to 16.26 (c) on page 192.
19.26

19.27
Unit 20 Problem Solutions

20.1  See FLD p. 743 for solution.

20.2  See FLD p. 743-744 for solution.

20.3  Replace line 12 with:

    signal State, NextState: integer range 0 to 5;

Replace lines 27 - 33 with:

    when 1 | 2 | 3 | 4 =>
      if M = '1' then Ad <= '1';
        Nextstate <= State;
      else Sh <= '1'; Nextstate <= State + 1; end if;
    when 5 =>
      Done <= '1'; Nextstate <= 0;

Replace lines 39 - 41 with:

    if Load = '1' then
      ACC <= "00000" & MPLIER; end if;
    if Ad = '1' then
      ACC(8 downto 4) <= addout; ACC(0) <= '0'; end if;

20.4  See FLD p. 744-745 for solution.

20.5  See FLD p. 745 for solution.

20.6  See FLD p. 746 for solution.

20.7  Replace line 14 with:

    signal Counter: integer range 0 to 4;
    signal State, NextState: integer range 0 to 3;

After line 22, add:

    if Counter = 3 then K <= '1' else '0';

Replace lines 33 - 36 with:

    when 2 =>
      if C = '1' then Su <= '1'; NextState <= 2;
      elsif K = '1' then Sh <= '1'; NextState <= 3;
      else Sh <= '1'; NextState <= 2; end if;
    when 3 =>
      if Sh = '1' then Dividend <= Dividend (7 downto 0) & '0';
        Counter <= Counter + 1; end if;

Replace line 47 with:

    if Sh = '1' then
      Dividend <= Dividend (7 downto 0) & '0';
      Counter <= Counter + 1; end if;

20.8  Entity and architecture for DiceGame goes here.

20.8 (contd)

    entity GameTest is
      end GameTest;
    architecture dicetest of GameTest is
      component DiceGame
        port (CLK, Rb, Reset : in bit;
            Sum: in integer range 2 to 12 ;
            Roll, Win, Lose: out bit);
      end component;
    signal rb, reset, clk, roll, win, lose: bit;
    signal sum: integer range 2 to 12;
    type arr is array(0 to 11) of integer;
    constant Sumarray:arr := (7,11,2,4,7,5,6,7,6,8,9,6);
    begin
      CLK <= not CLK after 20 ns;
      Dice: Dicegame port map(rb,reset,clk,sum,roll,win,lose);
      process
        begin
          for i in 0 to 11 loop
            Rb <= '1'; -- push roll button
            wait until roll = '1';
            wait until clk'event and clk = '1';
            Rb <= '0'; -- release roll button
            wait until roll = '0';
            sum <= Sumarray(i);
            -- read roll of dice from array
            wait until clk'event and clk = '1';
            wait until clk'event and clk = '1';
            if win = '1' or lose = '1' then reset <= '1';
            else if win = '1' then reset <= '0';
            wait until clk'event and clk = '1';
          end if;
          reset <= '0';
          end loop;
          -- test completed, do not execute process again
        end process;
      end dicetest;
    end architecture;
Replace lines 6 - 11 with:

```
Port (Dividend_in: in std_logic_vector(4 downto 0);
      Divisor: in std_logic_vector(4 downto 0);
      St, Clk: in std_logic;
      Quotient: out std_logic_vector(4 downto 0);
      Remainder: out std_logic_vector(4 downto 0);
```

Replace lines 14 - 17 with:
```
signal State, NextState: integer range 0 to 6;
signal C, Load, Su, Sh, V: std_logic;
signal Subout : std_logic_vector (5 downto 0);
signal Dividend: std_logic_vector (9 downto 0);
```

Replace lines 19 - 23 with:
```
Subout <= '0'&Dividend(9 downto 5) - Divisor;
C <= not Subout (5);
Remainder <= Dividend (9 downto 5);
V <= '1' when Divisor = "00000" else '0';
Quotient <= Dividend (4 downto 0);
State_Graph: process (State, St, C, V)
```

Replace line 25 with:
```
Load <= '0'; Sh <= '0'; Su <= '0';
```

Replace lines 28 - 33 with:
```
if (St = '1') then
  if (V='0') then Load <='1'; NextState <= 1;
  else Nextstate <= 0; end if;
else Nextstate <= 0; end if;
when 1 => Sh <='1'; NextState <= 2;
when 2 | 3 | 4 | 5 =>
```

Replace line 36 with:
```
when 6 =>
```

Replace lines 45 - 47 with:
```
if Load = '1' then Dividend <= "00000" & Dividend_in; end if;
if Su = '1' then Dividend(9 downto 5) <= Subout (4 downto 0); Dividend(0) <= '1'; end if;
if Sh = '1' then Dividend <= Dividend (8 downto 0) & '0'; end if;
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mult20_10 is
  Port (CLK, S: in std_logic;
        Mplier, Mcand : in std_logic_vector(3 downto 0);
        Product : out std_logic_vector(7 downto 0);
        D : out std_logic);
end mult20_10;

architecture Behavioral of mult20_10 is
  signal State, NextState: integer range 0 to 4;
  signal A, B: std_logic_vector (3 downto 0);    -- Multiplicand & Multiplier
  signal PU, PL: std_logic_vector (3 downto 0);   -- Product registers
  signal muxout, andarray: std_logic_vector (3 downto 0);
  signal addout: std_logic_vector (4 downto 0);
  signal BZ, LA, CP, DB, LPU, LPL, EA, MS, CC, C: std_logic;

begin
  BZ <= '1' when B = "0000" else '0';
  muxout <= PU when MS = '1' else PL;
  andarray <= A when EA = '1' else "0000";
  addout <= ('0' & muxout) + ('0' & andarray) + ("0000" & C);  -- adder output is
  Product <= PU & PL;                                                                            -- 5 bits including carry

  process (S, State, BZ)
  begin
    CP <= '0'; LA <= '0'; DB <= '0'; MS <= '0'; CC <= '0';
    EA <= '0'; LPU <= '0'; LPL <= '0'; D <= '0';          -- control signals are '0' by default
    case State is
      when 0 =>
        if S = '1' then NextState <= 1; else NextState <= 0; end if;
      when 1 =>
        if BZ = '1' then NextState <= 3; else LPL <= '1'; EA <= '1'; DB <= '1'; NextState <= 2; end if;
      when 2 =>
        LPU <= '1'; MS <= '1'; NextState <= 1;
      when 3 =>
        D <= '1';
        if S = '1' then NextState <= 3; else NextState <= 0; end if;
    end case;
  end process;

  process (CLK)
  begin
    if CLK'event and CLK = '1' then -- update registers on rising edge of clk
      if LA = '1' then B <= Mplier; A <= Mcand; end if; -- load multiplier & multiplicand
      if CP = '1' then PU <= "0000"; PL <= "0000"; end if; -- clear product registers
      if DB = '1' then B <= B - 1; end if; -- decrement multiplier
      if LPL = '1' then PL <= addout(3 downto 0); end if;
      if LPU = '1' then PU <= addout(3 downto 0); end if;
      if CC = '1' then C <= '0'; else C <= addout(4); end if -- load carry flip-flop
      State <= NextState;
    end if;
  end process;
end Behavioral;
Unit 20 Solutions

20.10 (b) library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
    use IEEE.STD_LOGIC_ARITH.ALL;
    use IEEE.STD_LOGIC_UNSIGNED.ALL;

    entity test20_10 is
    end test20_10;

    architecture test1 of test20_10 is
    component mult20_10
        port (Clk: in std_logic;
               S: in std_logic;
               Mplier, Mcand : in std_logic_vector(3 downto 0);
               Product : out std_logic_vector(7 downto 0);
               D: out std_logic);
    end component;

    constant N: integer := 4;
    type arr is array(1 to N) of std_logic_vector(3 downto 0);
    constant Mcandarr: arr := ("1011", "1011", "1111", "0000");
    constant Mplierarr: arr := ("0101", "0000", "1111", "1111");
    signal CLK: std_logic := '0';
    signal S, D: std_logic;
    signal Mplier, Mcand: std_logic_vector(3 downto 0);
    signal Product: std_logic_vector(7 downto 0);
    begin
        mult1: mult20_10 port map(CLk, S, Mplier, Mcand, Product, D);
        CLK <= not CLK after 10 ns;          -- clock has 20 ns period
        process
            for i in 1 to N loop
                Mcand <= Mcandarr(i);
                Mplier <= Mplierarr(i);
                S <= '1';
                wait until CLK = '1' and CLK'event;
                S <= '0';
                wait until D = '1';
                wait until CLK = '1' and CLK'event;
            end loop;
        end process;
    end test1;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mult20_11 is
  Port (CLK, S: in std_logic;
    Mplier, Mcand : in std_logic_vector(3 downto 0);
    Product : out std_logic_vector(7 downto 0);
    D : out std_logic);
end mult20_11;

architecture Behavioral of mult20_11 is
  signal State, NextState: integer range 0 to 4;
signal B: std_logic_vector (3 downto 0);  -- Multiplier counter
signal A: std_logic_vector (3 downto 0);  -- Multiplicand register
signal PU: std_logic_vector (3 downto 0);  -- Upper half of product register
signal PL: std_logic_vector (3 downto 0);  -- Lower half of product register
signal andarray: std_logic_vector (3 downto 0);
signal addout: std_logic_vector (4 downto 0);
signal muxout: std_logic_vector (3 downto 0);
signal BZ, LA, CP, DB, IB, IA, LPU, LPL, EA, MS, CC, SC, C: std_logic;
alias B3: std_logic is B(3);
alias A3: std_logic is A(3);
bEGIN  
  BZ <= '1' when B = "0000" else '0';
muxout <= PU when MS = '1' else PL;
andarray <= A when EA = '1' and IA = '0' else 
  not A when EA = '1' and IA = '1' else 
  "1111" when EA = '0' and IA = '1' else "0000";
addout <= ('0' & muxout) + ('0' & andarray) + ("0000" & C);  -- adder output is 5 bits
Product <= PU & PL;                                                                            -- including carry
process (S, State, BZ)
begin
  CP <= '0'; LA <= '0'; DB <= '0'; IB <= '0'; MS <= '0'; CC <= '0';     -- control signals are '0'
  SC <= '0'; EA <= '0'; IA <= '0'; LPU <= '0'; LPL <= '0'; D <= '0';    -- by default
  case State is
    when 0 =>
      CP <= '1'; LA <= '1';
      if S = '1' then NextState <= 1; else NextState <= 0; end if;
    when 1 =>
      NextState <= 2;
      if B3 = '1' then SC <= '1';
      else CC <= '1'; end if;
    when 2 =>
      if BZ = '1' then NextState <= 4;
      else NextState <= 3; LPL <= '1'; EA <= '1'; end if;
      if BZ = '0' and B3 = '1' then IA <= '1'; end if;
    when 3 =>
      LPU <= '1'; MS <= '1'; NextState <= 2;
      if B3 = '0' then CC <= '1'; DB <= '1';
      else SC <= '1'; IB <= '1'; end if;
      if (A3 xor B3) = '1' then IA <= '1'; end if;
    when 4 =>
      D <= '1';
      if S = '1' then NextState <= 4;
      else NextState <= 0; end if;
  end case;
end process;
Unit 20 Solutions

20.11 (a) process (CLK)
begin
if CLK'event and CLK = '1' then -- update registers on rising edge of clk
    if LA = '1' then B <= Mplier;
        A <= Mcand; end if; -- load multiplier & multiplicand
    if CP = '1' then PU <= "0000";
        PL <= "0000"; end if; -- clear product registers
    if DB = '1' then B <= B - 1; end if; -- decrement multiplier
    if IB = '1' then B <= B + 1; end if; -- increment multiplier
    if LPL = '1' then PL <= addout(3 downto 0); end if;
    if LPU = '1' then PU <= addout(3 downto 0); end if;
    elsif SC = '1' then C <= '1';
        else C <= addout(4); end if; -- load carry flip-flop
    State <= NextState;
end if;
end process;
end Behavioral;

20.11 (b)
-- Same as 20-10(b) except
constant N: integer := 6;
type arr is array(1 to N) of std_logic_vector(3 downto 0);
constant Mcandarr: arr := ("0111", "0000", "0111", "1000", "0111", "1000");
constant Mplierarr: arr := ("0111", "0111", "0000", "0111", "1000", "1000");