



This material is protected by U.S. laws. And can only be provided to instructors. MyUET is providing this material specifically for instructors of University of Engineering and Technology Lahore. myUET is not responsible for any misuse, illegal Publishing or any illegal activity related to this material.

⚠ This work is protected by US copyright laws and is for instructors' use only.

Instructor's Resource Manual
to accompany

DIGITAL SYSTEMS

Principles and Applications

Tenth Edition

Ronald J. Tocci
Neal S. Widmer
Gregory L. Moss

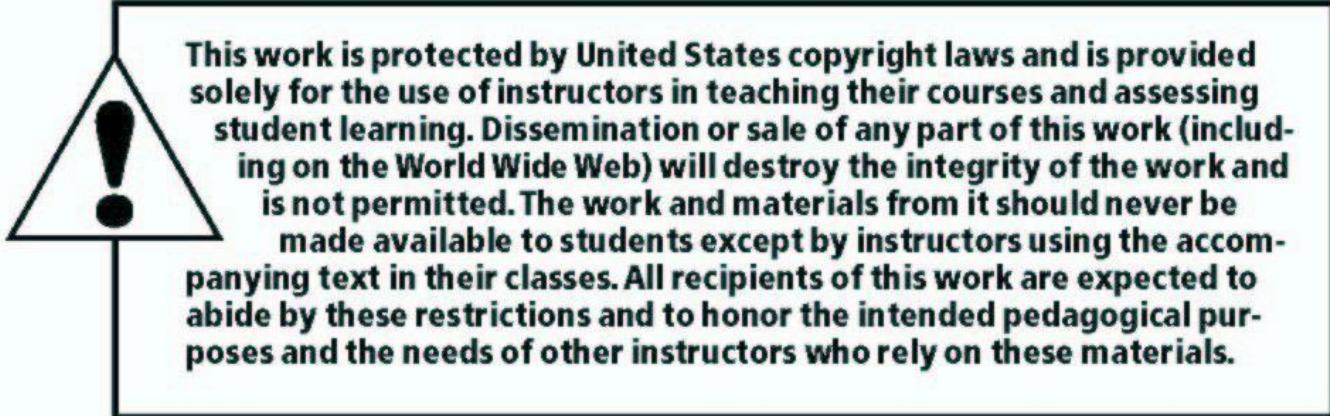
Prepared by
Frank J. Ambrosio
Monroe Community College

my  UET

PEARSON

Prentice Hall

Upper Saddle River, New Jersey
Columbus, Ohio



Copyright © 2007 by Pearson Education, Inc., Upper Saddle River, New Jersey 07458.
Pearson Prentice Hall. All rights reserved. Printed in the United States of America. This publication is
protected by Copyright and permission should be obtained from the publisher prior to any prohibited
reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic,
mechanical, photocopying, recording, or likewise. For information regarding permission(s), write to: Rights
and Permissions Department.

Pearson Prentice Hall™ is a trademark of Pearson Education, Inc.

Pearson® is a registered trademark of Pearson plc

Prentice Hall® is a registered trademark of Pearson Education, Inc.

Instructors of classes using Tocci, Widmer, and Moss, Digital Systems: Principles and Applications, 10th
Edition, may reproduce material from the instructor's manual for classroom use.

10 9 8 7 6 5 4 3 2 1



ISBN 0-13-172665-X

TABLE OF CONTENTS

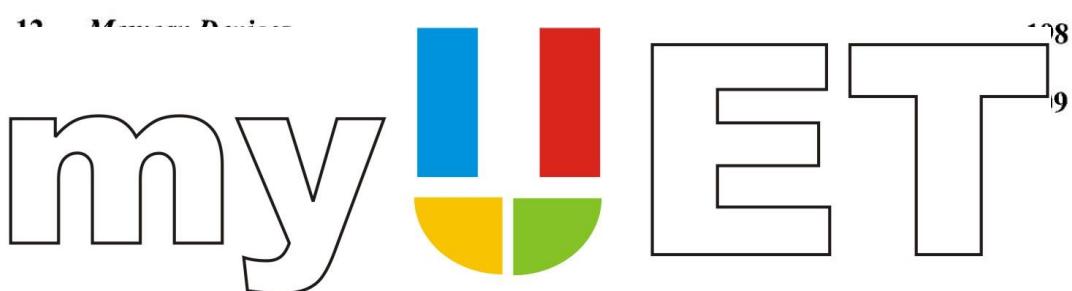
SOLUTIONS TO:

RONALD J. TOCCI/NEAL S. WIDMER/GREGORY L. MOSS

DIGITAL SYSTEMS: PRINCIPLES AND APPLICATIONS

TENTH EDITION

CHAPTER 1	<i>Introductory Concepts</i>	1
CHAPTER 2	<i>Number Systems and Codes</i>	3
CHAPTER 3	<i>Describing Logic Circuits</i>	9
CHAPTER 4	<i>Combinational Logic Circuits</i>	29
CHAPTER 5	<i>Flip-Flops and Related Devices</i>	54
CHAPTER 6	<i>Digital Arithmetic: Operations and Circuits</i>	80
CHAPTER 7	<i>Counters and Registers</i>	98
CHAPTER 8	<i>Integrated-Circuit Logic Families</i>	147
CHAPTER 9	<i>MSI Logic Circuits</i>	156
CHAPTER 10	<i>Digital System Projects Using HDL</i>	178
CHAPTER 11	<i>Interfacing with the Analog World</i>	189
CHAPTER 12	<i>Appendix A: Boolean Algebra</i>	198
CHAPTER 13	<i>Appendix B: Minimization of Logic Functions</i>	199



CHAPTER ONE - Introductory Concepts

1.1 (a), (e) are digital.
 (b), (c), (d) are analog.

1.2 (a) Analog
 (b) Analog
 (c) Digital
 (d) Digital
 (e) Analog

1.3 (a) $11001_2 = 16+8+1 = 25_{10}$
 (b) $1001.1001_2 = 8+1+0.5+0.0625 = 9.5625_{10}$
 (c) $10011011001.10110_2 = 1024+128+64+16+8+1+0.5+0.125+0.0625 = 1241.6875_{10}$

1.4 (a) $10011_2 = 16+2+1 = 19_{10}$
 (b) $1100.0101_2 = 8+4+0.25+0.0625 = 12.3125_{10}$
 (c) $10011100100.10010_2 = 1252.5625_{10}$

1.5 $000_2, 001_2, 010_2, 011_2, 100_2, 101_2, 110_2, 111_2$

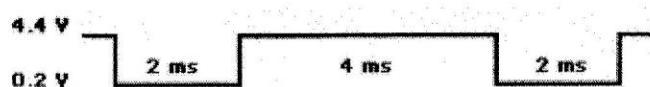
1.7 $2^{N-1} = 2^{10-1} = 1023$

1.8 $2^{N-1} = 2^{14-1} = 16,383$

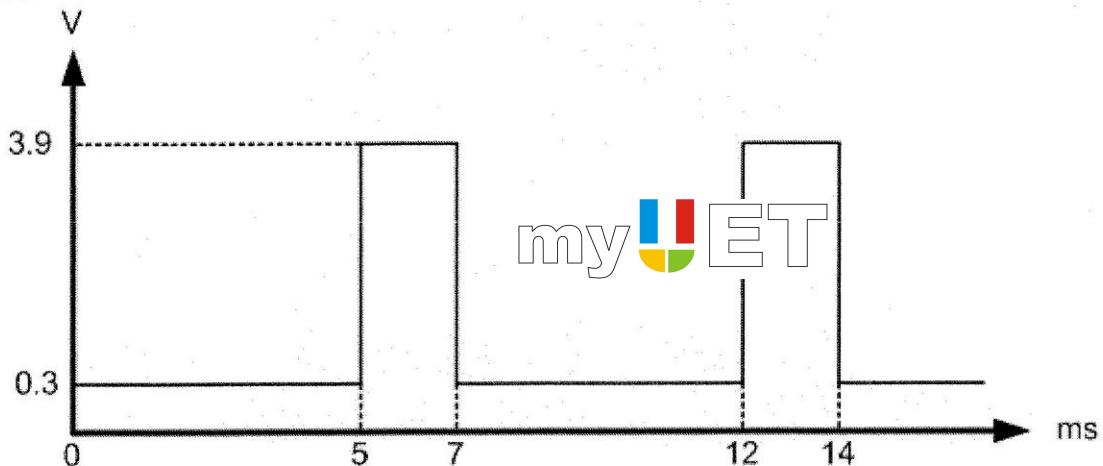
1.9 $2^8 = 256$ and $2^9 = 512$, therefore **9 bits are needed.**

1.10 $2^{N-1} = 63$, therefore **6 bits are needed.**

1.11



1.12



1.13 (a) $2^{N-1} = 15$

N = 4; Therefore, **4 lines** are required for parallel transmission.

(b) Only **1 line** is required for serial transmission.

1.14 A **microprocessor** is a CPU on a chip. The CPU contains the control unit and the arithmetic logic unit (ALU). A **microcomputer** generally consists of several IC chips including a microprocessor chip, memory chips, and input/output interface chips along with input/output devices.

1.15 A **microcontroller** is a specialized type of microcomputer that is designed to be used as a dedicated or embedded controller. Microcontrollers are generally much smaller than general-purpose microcomputers.



CHAPTER TWO - Number Systems and Codes

- 2.1** (a) $10110_2 = 16+4+2 = 22_{10}$
 (b) $10010101_2 = 128+16+4+1 = 149_{10}$
 (c) $100100001001_2 = 2048+256+8+1 = 2313_{10}$
 (d) $01101011_2 = 64+32+8+2+1 = 107_{10}$
 (e) $11111111_2 = 128+64+32+16+8+4+2+1 = 255_{10}$
 (f) $01101111_2 = 64+32+8+4+2+1 = 111_{10}$
 (g) $1111010111_2 = 512+256+128+64+16+4+2+1 = 983_{10}$
 (h) $11011111_2 = 128+64+16+8+4+2+1 = 223_{10}$

- 2.2** (a) $37_{10} = 32+4+1 = 100101_2$
 (b) $13_{10} = 8+4+1 = 1101_2$
 (c) $189_{10} = 128+32+16+8+4+1 = 10111101_2$
 (d) $1000_{10} = 512+256+128+64+32+8 = 1111101000_2$
 (e) $77_{10} = 64+8+4+1 = 1001101_2$
 (f) $390_{10} = 256+128+4+2 = 110000110_2$
 (g) $205_{10} = 128+64+8+4+1 = 11001101_2$
 (h) $2133_{10} = 2048+64+16+4+1 = 100001010101_2$
 (i) $511_{10} = 256+128+64+32+16+8+4+2+1 = 11111111_2$

2.3 $(2^8-1) = 255_{10}; (2^{16}-1) = 65,535_{10}$

- 2.4** (a) $743_{16} = 7 \times 16^2 + 4 \times 16^1 + 3 \times 16^0 = 1859_{10}$
 (b) $36_{16} = 3 \times 16^1 + 6 \times 16^0 = 54_{10}$
 (c) $37FD_{16} = 3 \times 16^3 + 7 \times 16^2 + 15 \times 16^1 + 13 \times 16^0 = 14333_{10}$
 (d) $2000_{16} = 2 \times 16^3 = 8192_{10}$
 (e) $165_{16} = 1 \times 16^2 + 6 \times 16^1 + 5 \times 16^0 = 357_{10}$
 (f) $ABCD_{16} = 10 \times 16^3 + 11 \times 16^2 + 12 \times 16^1 + 13 \times 16^0 = 43981_{10}$
 (g) $7FF_{16} = 7 \times 16^2 + 15 \times 16^1 + 15 \times 16^0 = 2047_{10}$
 (h) $1204_{16} = 1 \times 16^3 + 2 \times 16^2 + 4 \times 16^0 = 4612_{10}$

- 2.5** (a) $\begin{array}{rcl} 59/16 &= 3 & \text{Remainder of } 11 \\ 3/16 &= 0 & \text{Remainder of } 3 \end{array} \} 59_{10} = 3B_{16}$
 (b) $\begin{array}{rcl} 372/16 &= 23 & \text{Remainder of } 4 \\ 23/16 &= 1 & \text{Remainder of } 7 \\ 1/16 &= 0 & \text{Remainder of } 1 \end{array} \} 372_{10} = 174_{16}$
 (c) $\begin{array}{rcl} 919/16 &= 57 & \text{Remainder of } 7 \\ 57/16 &= 3 & \text{Remainder of } 9 \\ 3/16 &= 0 & \text{Remainder of } 3 \end{array} \} 919_{10} = 397_{16}$



(d) $1024/16 = 64$ Remainder of 0 }
 $64/16 = 4$ Remainder of 0 }
 $4/16 = 0$ Remainder of 4 } $1024_{10} = 400_{16}$

(e) $771/16 = 48$ Remainder of 3 }
 $48/16 = 3$ Remainder of 0 }
 $3/16 = 0$ Remainder of 3 } $771_{10} = 303_{16}$

(f) $2313/16 = 144$ Remainder of 9 }
 $144/16 = 9$ Remainder of 0 }
 $9/16 = 0$ Remainder of 9 } $2313_{10} = 909_{16}$

(g) $65536/16 = 4096$ Remainder of 0 }
 $4096/16 = 256$ Remainder of 0 }
 $256/16 = 16$ Remainder of 0 }
 $16/16 = 1$ Remainder of 0 }
 $1/16 = 0$ Remainder of 1 } $65,536_{10} = 10000_{16}$

(h) $255/16 = 15$ Remainder of 15 }
 $15/16 = 0$ Remainder of 15 } $255_{10} = FF_{16}$

- 2.6** (a) $743_{16} = 11101000011_2$ (b) $36_{16} = 110110_2$
 (c) $37FD_{16} = 1101111111101_2$ (d) $2000_{16} = 10000000000000_2$
 (e) $165_{16} = 101100101_2$ (f) $ABCD_{16} = 101010111001101_2$ (g) $7FF_{16} = 01111111111_2$
 (h) $1204_{16} = 1001000000100_2$

- 2.7** (a) $10110_2 = 16_{16}$ (b) $10010101_2 = 95_{16}$ (c) $100100001001_2 = 909_{16}$ (d) $01101011_2 = 6B_{16}$
 (e) $11111111_2 = FF_{16}$ (f) $01101111_2 = 6F_{16}$ (g) $1111010111_2 = 3D7_{16}$ (h) $11011111_2 = DF_{16}$

- 2.8** 195, 196, 197, 198, 199, 19A, 19B, 19C, 19D, 19E, 19F, 2A0, 2A1, 2A2,..., 2A9, 2AA, 2AB,
 2AC, 2AD, 2AE, 2AF, 2B0.

2.9 $2133/16 = 133$ Remainder of 5 }
 $133/16 = 8$ Remainder of 5 }
 $8/16 = 0$ Remainder of 8 } $2133_{10} = 855_{16} = 100001010101_2$

2.10 $16^N \geq 20,000$; Therefore, n=4

- 2.11** (a) $92_{16} = 9 \times 16^1 + 2 \times 16^0 = 146_{10}$ (b) $1A6_{16} = 1 \times 16^2 + 10 \times 16^1 + 6 \times 16^0 = 422_{10}$
 (c) $37FD_{16} = 3 \times 16^3 + 7 \times 16^2 + 15 \times 16^1 + 13 \times 16^0 = 14333_{10}$
 (d) $ABCD_{16} = 10 \times 16^3 + 11 \times 16^2 + 12 \times 16^1 + 13 \times 16^0 = 43981_{10}$
 (e) $000F_{16} = 0 \times 16^3 + 0 \times 16^2 + 0 \times 16^1 + 15 \times 16^0 = 15_{10}$
 (f) $55_{16} = 5 \times 16^1 + 5 \times 16^0 = 85_{10}$ (g) $2C0_{16} = 2 \times 16^2 + 12 \times 16^1 + 0 = 704_{10}$
 (h) $7FF_{16} = 7 \times 16^2 + 15 \times 16^1 + 15 \times 16^0 = 2455_{10}$



2.12	(a)	75/16	= 4	Remainder of 11 [B]	}	
		4/16	= 0	Remainder of 4	}	$75_{10} = 4B_{16}$
	(b)	314/16	= 19	Remainder of 10 [A]	}	
		19/16	= 1	Remainder of 3	}	
		1/16	= 0	Remainder of 1	}	$314_{10} = 13A_{16}$
	(c)	2048/16	= 128	Remainder of 0	}	
		128/16	= 8	Remainder of 0	}	
		8/16	= 0	Remainder of 8	}	$2048_{10} = 800_{16}$
	(d)	24/16	= 1	Remainder of 8		
		1/16	= 0	Remainder of 1		$24_{10} = 18_{16}$
	(e)	7245/16	= 452	Remainder of 13 [D]	}	
		452/16	= 28	Remainder of 4	}	
		28/16	= 1	Remainder of 12 [C]	}	
		1/16	= 0	Remainder of 1	}	$7245_{10} = 1C4D_{16}$
	(f)	498/16	= 31	Remainder of 2	}	
		31/16	= 1	Remainder of 15 [F]	}	
		1/16	= 0	Remainder of 1		$498_{10} = 1F2_{16}$
	(g)	25619/16	= 1601	Remainder of 3	}	
		1601/16	= 100	Remainder of 1	}	
		100/16	= 6	Remainder of 4	}	
		6/16	= 0	Remainder of 6		$25619_{10} = 6413_{16}$
	(h)	4095/16	= 255	Remainder of 15 [F]	}	
		255/16	= 15	Remainder of 15 [F]	}	
		15/16	= 0	Remainder of 15 [F]		$4095_{10} = FFF_{16}$

- 2.13 (a) 9 (b) D (c) 8 (d) 0 (e) F (f) 2 (g) A (h) 9 (i) B (j) C (k) 3
 (l) 4 (m) 1 (n) 5 (o) 7 (p) 6

- 2.14 (a) 0110 (b) 0111 (c) 0101 (d) 0001 (e) 0100 (f) 0011 (g) 1100 (h) 1011
 (i) 1001 (j) 1010 (k) 0010 (l) 1111 (m) 0000 (n) 1000 (o) 1101 (p) 1001

- 2.15 (a) $00010110_2 = 16_{16}$ (b) $10010101_2 = 95_{16}$ (c) $100100001001_2 = 909_{16}$ (d) $01101011_2 = 6B_{16}$
 (e) $11111111_2 = FF_{16}$ (f) $01101111_2 = 6F_{16}$ (g) $001111010111_2 = 3D7_{16}$ (h) $11011111_2 = DF_{16}$

- 2.16 (a) $92_{16} = 10010010_2$ (b) $1A6_{16} = 000110100110_2$ (c) $37FD_{16} = 001101111111101_2$
 (d) $ABCD_{16} = 1010101111001101_2$ (e) $000F_{16} = 1111_2$ (f) $55_{16} = 01010101_2$
 (d) $2C0_{16} = 001011000000_2$ (e) $7FF_{16} = 011111111111_2$

- 2.17 $280_{16}, 281_{16}, 282_{16}, \dots, 288_{16}, 289_{16}, 28A_{16}, 28B_{16}, 28C_{16}, 28D_{16}, 28E_{16}, 28F_{16}, 290_{16},$
 $291_{16}, \dots, 298_{16}, 299_{16}, 29A_{16}, 29B_{16}, 29C_{16}, 29D_{16}, 29E_{16}, 29F_{16}, 2A0_{16}$

- 2.18 With *four* hex digits we can represent a decimal number up to: $FFFF_{16} = (16^4 - 1) = 65,535_{10}$
 With *five* hex digits we can represent a decimal number up to: $FFFFFF_{16} = (16^5 - 1) = 1,048,575_{10}$
 Therefore, we need five hex digits to represent decimal numbers up to 1 million.

- 2.19 (a) $47_{10} = 0100\ 0111_{BCD}$ (b) $962_{10} = 1001\ 0110\ 0010_{BCD}$ (c) $187_{10} = 0001\ 1000\ 0111_{BCD}$
 (d) $6727_{10} = 0110\ 0111\ 0010\ 0111_{BCD}$ (e) $13_{10} = 0001\ 0011_{BCD}$
 (f) $529_{10} = 0101\ 0010\ 1001_{BCD}$ (g) $89,627_{10} = 1000\ 1001\ 0110\ 0010\ 0111_{BCD}$
 (h) $1024_{10} = 0001\ 0000\ 0010\ 0100_{BCD}$

- 2.20 (a) $(2^{N-1}) = 999$. Therefore, $N=10$. Hence, it requires **10 bits** for straight binary.
 (b) 999_{10} requires **12 bits** for BCD (4 bits per digit).

- 2.21 (a) $1001\ 0111\ 0101\ 0010_{BCD} = 9752_{10}$ (b) $0001\ 1000\ 0100_{BCD} = 184_{10}$
 (c) $0110\ 1001\ 0101_{BCD} = 695_{10}$ (d) $0111\ 0111\ 0111\ 0101_{BCD} = 7775_{10}$
 (e) $0100\ 1001\ 0010_{BCD} = 492_{10}$ (f) $0101\ 0101\ 0101_{BCD} = 555_{10}$

- 2.22 (a) 1 byte = 8 bits. Thus, 8 bytes = **64 bits**
 (b) 4 bytes = 32 bits. A hex digit requires four bits to be represented. Thus, the largest hex number that can be represented in four bytes is **FFFFFFFFFF₁₆**.
 (c) The largest BCD-encoded decimal value that can be represented in three bytes is **999,999**.

- 2.23 (a) 0101 (b) 4 nibbles (c) 3 bytes

2.24 **x = 3*y** **Hex** **Bin** **With odd-parity**

x ----->	78 = 111 1000	1111 1000 = F8
space ----->	20 = 010 0000	0010 0000 = 20
= ----->	3D = 011 1101	0011 1101 = 3D
space ----->	20 = 010 0000	0010 0000 = 20
3 ----->	3 = 011 0011	1011 0011 = B3
* ----->	2A = 010 1010	0010 1010 = 2A
y ----->	79 = 111 1001	0111 1001 = 79

1111 1000 0010 0000 0011 1101 0010 0000 1011 0011 0010 1010 0111 1001
 x space = space 3 * y

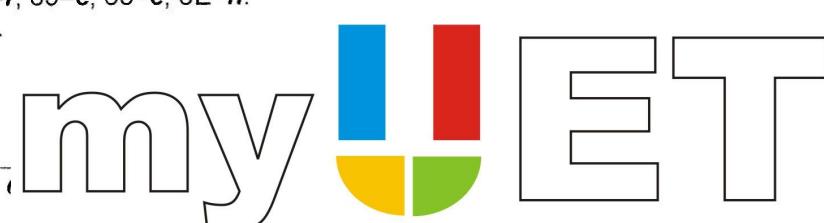
2.25 **x = 3*y** **Hex** **Bin** **With even-parity**

x ----->	78 = 111 1000	0111 1000 = 78
space ----->	20 = 010 0000	1010 0000 = A0
= ----->	3D = 011 1101	1011 1101 = BD
space ----->	20 = 010 0000	1010 0000 = A0
3 ----->	3 = 011 0011	0011 0011 = 33
* ----->	2A = 010 1010	1010 1010 = AA
y ----->	79 = 111 1001	1111 1001 = F9

1111 1000 0010 0000 0011 1101 0010 0000 1011 0011 0010 1010 0111 1001
 x space = space 3 * y

- 2.26 (a) 42=B; 45=E; 4E=N; 20=blank; 53=S; 4D=M; 49=I; 54=T; 48=H.
 Thus, the name of the person is **BEN SMITH**.

- (b) 4A=J; 6F=o; 65=e; 20=blank; 47=G; 72=r; 65=e; 6E=n.
 Thus, the name of the person is **Joe Green**.



- 2.27** (a) $74_{10} = 01110100_{BCD}$ } 101110100 (b) $38_{10} = 00111000_{BCD}$ } 000111000
 (c) $8884_{10} = 100010001000100_{BCD}$ } 1100010001000100
 (d) $275_{10} = 001001110101_{BCD}$ } 0001001110101
 (e) $165_{10} = 000101100101_{BCD}$ } 0000101100101
 (f) $9201_{10} = 1001001000000001_{BCD}$ } 11001001000000001

- 2.28** (a) $\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{array}$ { parity bit
 9 5 8

Since the number of 1s is 5, there is **no single-bit** error.

- (b) $\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{array}$
 4 7 6

Since there are six 1s there is a **single** error.

- c) $\begin{array}{ccc} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{array}$
 7 12 1

There are seven 1s. However, the second BCD code group has an error since 1100 is an illegal BCD code. Thus, there must be a **double** error because there are an odd number of 1s.

- (d) $\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array}$
 8 6 2

There are five 1s. Thus, **no single-bit** errors.

- 2.29** 01001000 } O.K
 11000101 } O.K
 11001100 } O.K

11001000 } There is a single error.

11001100 } Error can't be detected by the receiver.



- 2.30** (a) 10110001001_2 (b) 11111111_2 (c) 209_{10} (d) $59,943_{10}$ (e) $9C1_{16}$ (f) 010100010001_{BCD}
 (g) 565_{10} (h) $10DC_{16}$ (i) 1961_{10} (j) $15,900_{10}$ (k) 640_{16} (l) $952B_{16}$
 (m) 100001100101_{BCD} (n) 947_{10} (o) 10001100101_2 (p) 101100110100_2
 (q) Convert to decimal, then to binary to obtain 1001010_2
 (r) Convert to decimal, then to BCD to obtain 01011000_{BCD}

- 2.31** (a) 100101_2 (b) 00110111_{BCD} (c) 25_{16} (d) 01100110110111_{ASCII}

- 2.32** (a) Hex (b) Two (c) digit (d) Gray code (e) parity bit/errors (f) ASCII (g) Hex (h) Byte

- 2.33** (a) 1000_2 (b) 010100_2 (c) 1100_2

- 2.34** (a) 1011_2 (b) 100111_2 (c) 1101_2

- 2.35** (a) $777A_{16}$ (b) $999A_{16}$ (c) 1000_{16} (d) 2001_{16} (e) $A00_{16}$ (f) $100B_{16}$

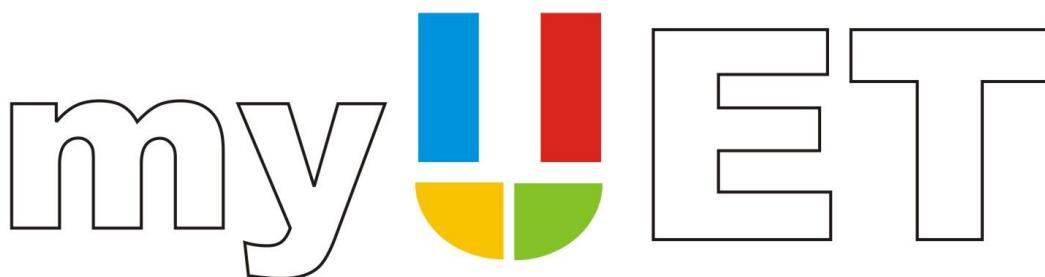
- 2.36** (a) 7778_{16} (b) 9998_{16} (c) $0FFE_{16}$ (d) $1FFF_{16}$ (e) $9FE_{16}$ (f) 1009_{16}

- 2.37** (a) A 20-bit address will allow $1,048,576$ (2^{20}) different memory locations to exist.
 (b) Since a hex digit requires 4 bits to represent, it will take 5 hex digits to represent the 20-bit address of a memory location.
 (c) $000FF_{16}$

- 2.38** (a) $2^6=64$ different voltage values; $2^8=256$ different voltage values; $2^{10}=1,024$ different voltage values.
(b) In 1s there are about 44,000 samples of 10-bits each recorded on the CD surface. Thus, there are about 440,000 bits recorded on the CD disk during 1s of sampling.
(c) There are about 440,000 bits recorded on the CD disk in 1 second of audio. Therefore, 5 billion bits of audio stored on the CD disk will be equivalent to approximately 11,363.63 seconds ($5 \times 10^9 / 440,000$).

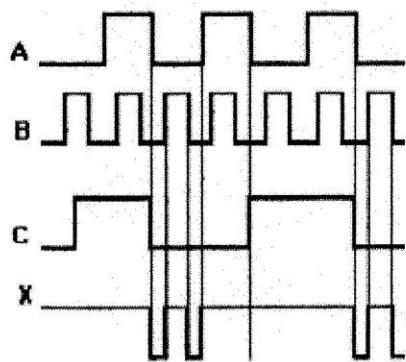
2.39 $254=2^x$. Therefore $x=7.98\approx 8$ -bits

2.40 Mega = $2^{20} = 1,048,576$
3 Bytes/pixel (1 byte per primary color)
 $(3 \text{ Bytes/pixel}) \times 3 \times 1,048,576 = 9,437,184 \text{ Bytes/photo}$
Memory card capacity = $128 \times 1,048,576 = 134,217,728 \text{ Bytes/card}$
Thus, $(134,217,728 \text{ Bytes/card}) / (9,437,184 \text{ Bytes/photo}) = 14.2 \text{ photos/card or } \underline{\text{14 Pictures.}}$

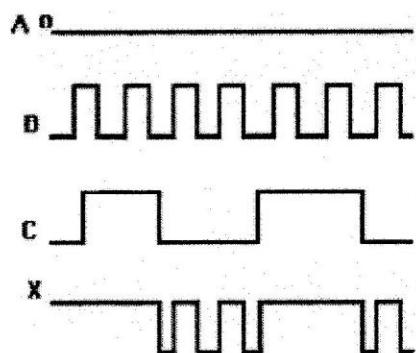


CHAPTER THREE - Describing Logic Circuits

3.1

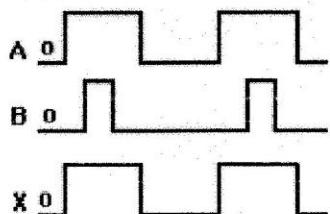


3.2

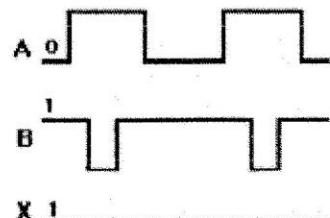


3.3 With A=1, X will always be 1 since the OR gate output is 1 whenever any input is a 1.

3.4 (a) Here's one case that refutes this statement.

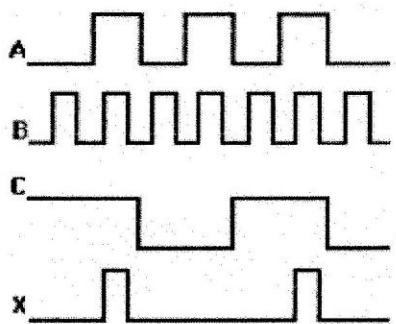


(b) Here's one case that refutes this statement.



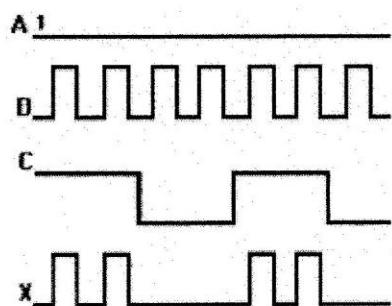
- 3.5 There are $2^5=32$ different input conditions. Only one of these (the 00000 condition) produces a LOW output.

- 3.6 (a)



(b) X = constant LOW.

(c)



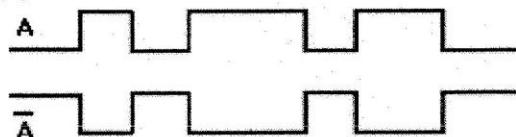
- 3.7 Change the OR gate to an AND gate.

- 3.8 OUT is always LOW since one or more inputs is always LOW.

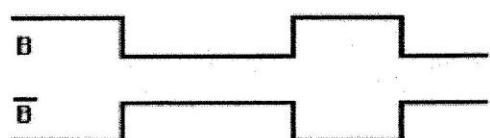
- 3.9 A logic HIGH and a logic LOW applied to the inputs of the unknown 2-input gate would tell us what type of gate it is. If the resulting output logic level is HIGH, then the gate is an OR gate. If the resulting output logic level is LOW, then the gate is an AND gate.

- 3.10 True. The output of any AND gate will be HIGH only when all of its inputs are HIGH.

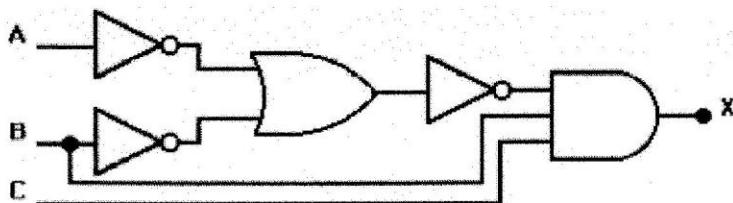
- 3.11 (a)



(b)



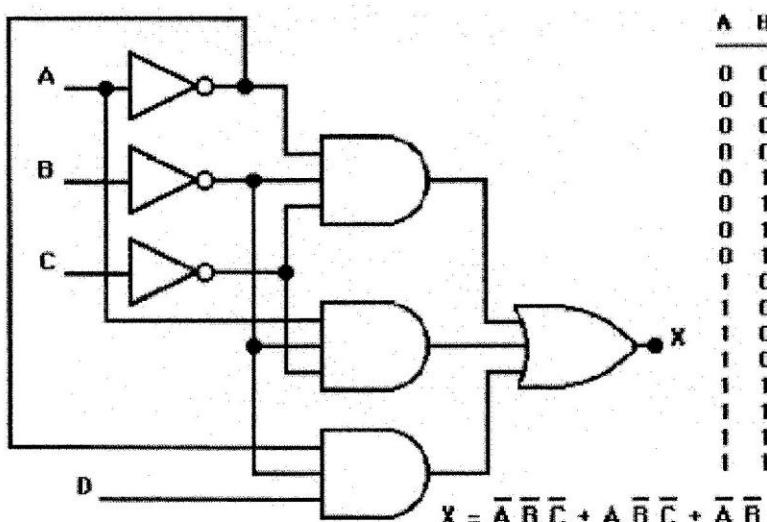
3.12 (a)



A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$X = [\bar{A} + \bar{B}] \cdot BC$

(b)



A	B	C	D	X
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}D$

3.13

E	D	C	B	A	$(A+B)$	$(A+B)C$	$[I(A+B)C]'$	$D+[I(A+B)C]'$	$[D+((A+B)C)']E$
0	0	0	0	0	0	0	1	1	0
0	0	0	0	1	1	0	1	1	0
0	0	0	1	0	1	0	1	1	0
0	0	0	1	1	1	0	1	1	0
0	0	1	0	0	0	0	1	1	0
0	0	1	0	1	1	1	0	0	0
0	0	1	1	0	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	1	0	0	0	0	0	1	1	0
0	1	0	0	1	1	0	1	1	0
0	1	0	1	0	1	0	1	1	0
0	1	0	1	1	1	0	1	1	0
0	1	1	0	0	0	0	1	1	0
0	1	1	0	1	1	1	0	1	0
0	1	1	1	0	1	1	0	1	0
1	0	0	0	0	0	0	1	1	1
1	0	0	0	1	1	0	1	1	1
1	0	0	1	1	1	0	1	1	1
1	0	1	0	0	0	0	1	1	1
1	0	1	0	1	1	1	0	0	0
1	0	1	1	0	1	1	0	0	0
1	0	1	1	1	1	1	0	1	1
1	1	0	0	0	0	0	1	1	1
1	1	0	0	1	1	0	1	1	1
1	1	0	1	0	1	0	1	1	1
1	1	0	1	1	1	0	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	0	1	1	1	0	1	1
1	1	1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	0	1	1

3.14

E	D	C	B	A	AB	$(AB)+C$	$[I(AB)+C]'$	$D[I(AB)+C]'$	$D[I(AB)+C]'+E$
0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	0	1	0	0	0
0	0	1	0	1	0	1	0	0	0
0	0	1	1	0	0	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	1	0	0	0	0	0	1	1	1
0	1	0	0	1	0	0	1	1	1
0	1	0	1	0	0	0	1	1	1
0	1	0	1	1	1	0	1	1	1
0	1	1	0	0	0	0	1	0	0
0	1	1	0	1	1	1	0	0	0
0	1	1	1	0	0	1	0	0	0
0	1	1	1	1	1	1	0	0	0

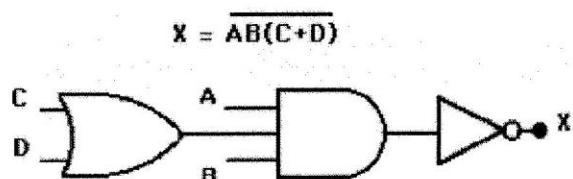


<i>E</i>	<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>	<i>AB</i>	$(AB)+C$	$\overline{[(AB)+C]'$	$D\overline{[(AB)+C]'$	$D\overline{[(AB)+C]'+E}$
1	0	0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	1	0	1
1	0	0	1	0	0	0	1	0	1
1	0	0	1	1	1	1	0	0	1
1	0	1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	0	0	1
1	0	1	1	0	0	1	0	0	1
1	0	1	1	1	1	1	0	0	1
1	1	0	0	0	0	0	1	1	1
1	1	0	0	1	0	0	1	1	1
1	1	0	1	0	0	0	1	1	1
1	1	0	1	1	1	1	0	0	1
1	1	1	0	0	0	1	0	0	1
1	1	1	0	1	0	1	0	0	1
1	1	1	1	0	0	1	0	0	1
1	1	1	1	1	1	1	0	0	1

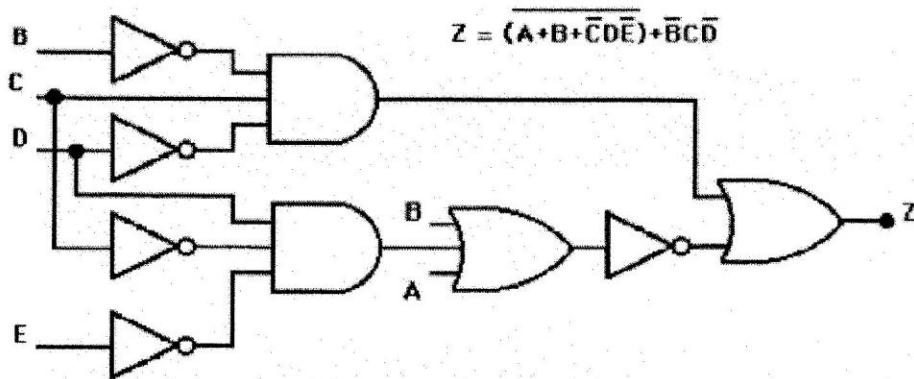
3.15

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	$A'BC$	$A+D$	$(A+D)'$	$(A+D)'(A'BC)$
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	1
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	0
1	1	0	0	0	1	0	0
1	1	0	1	0	1	0	0
1	1	1	0	0	1	0	0
1	1	1	1	0	1	0	0

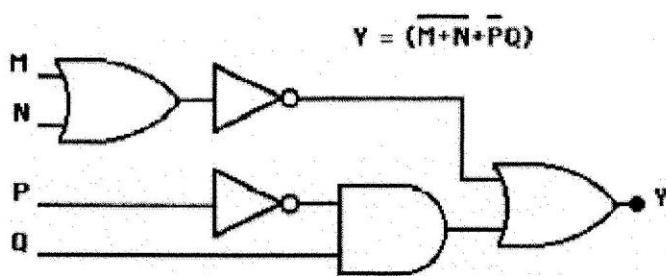
3.16 (a)



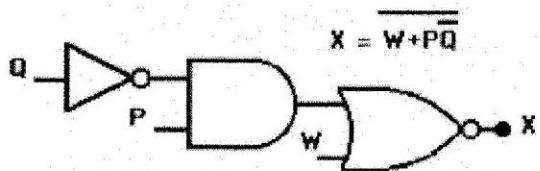
(b)



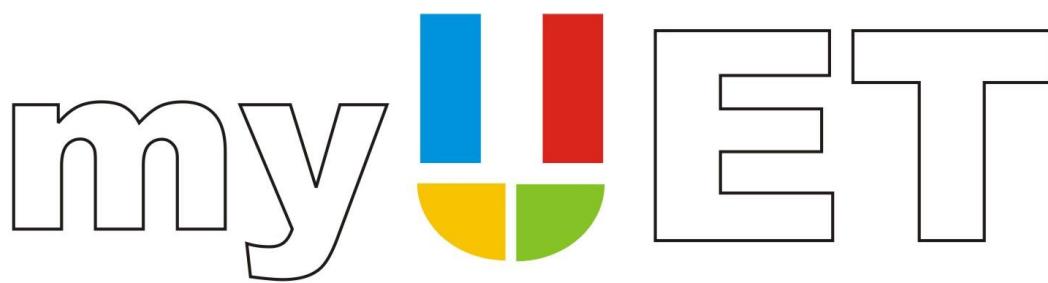
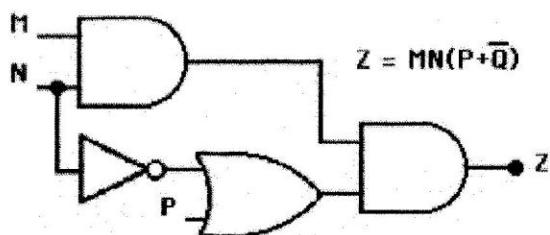
(c)



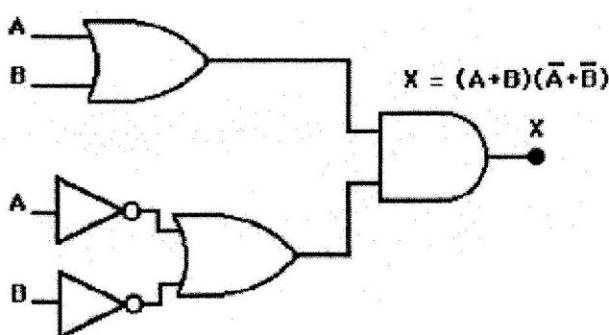
(d)



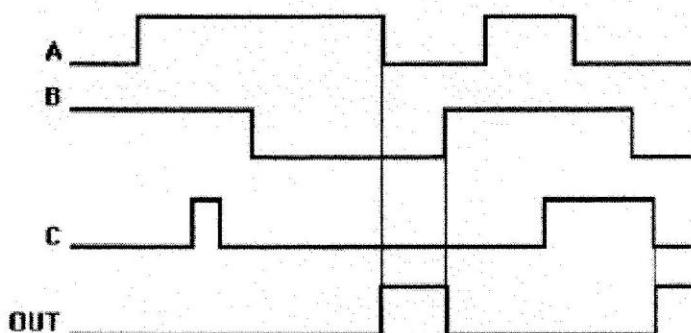
(e)



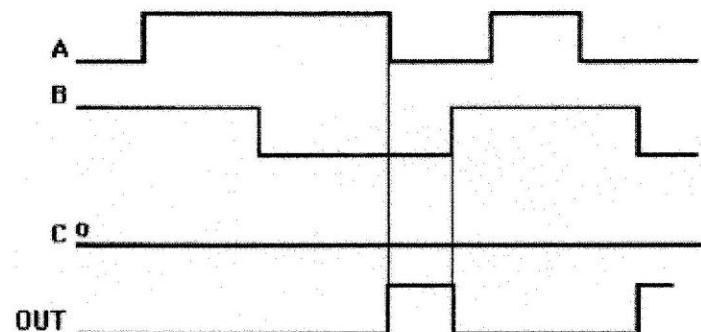
(f)



3.17 (a) OUT = 1 only when all inputs are = 0.

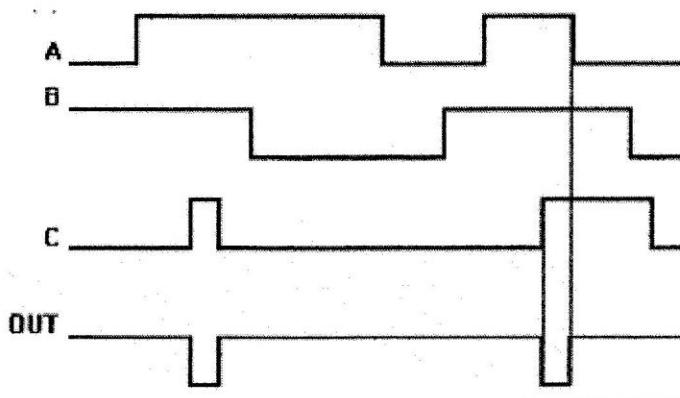


(b) With C = 0

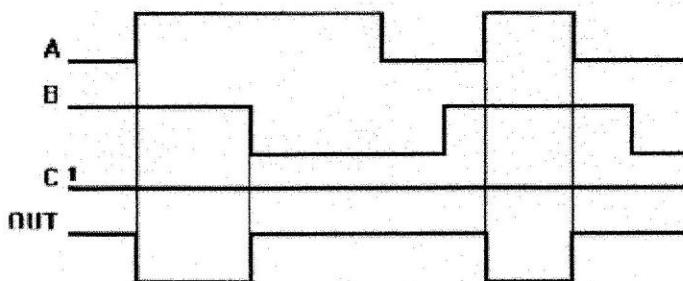


(c) With C = 1, OUT = 0 at all times.

3.18 (a) OUT = 0 only when all inputs are = 1.



- (b) With C = 0, OUT = 1 at all times.
 (c) With C = 1

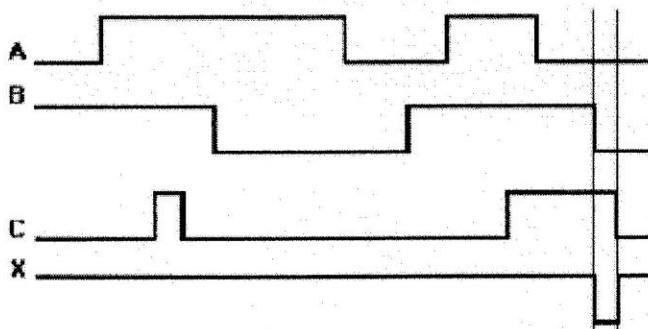


3.19

$$X = \overline{(A+B)}(B+\overline{C})$$

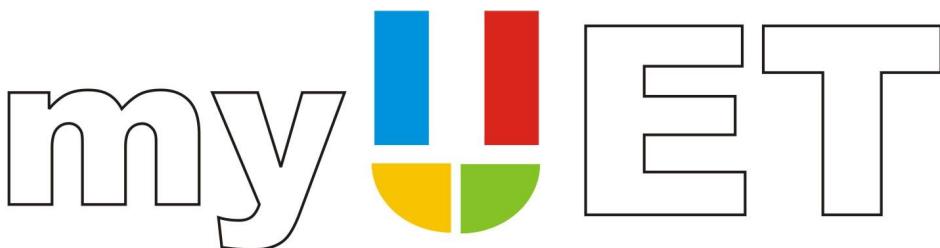
$$X = (A+B) + (B+\overline{C})$$

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

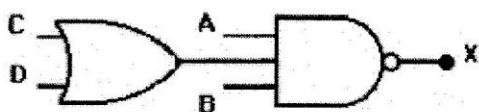


3.20

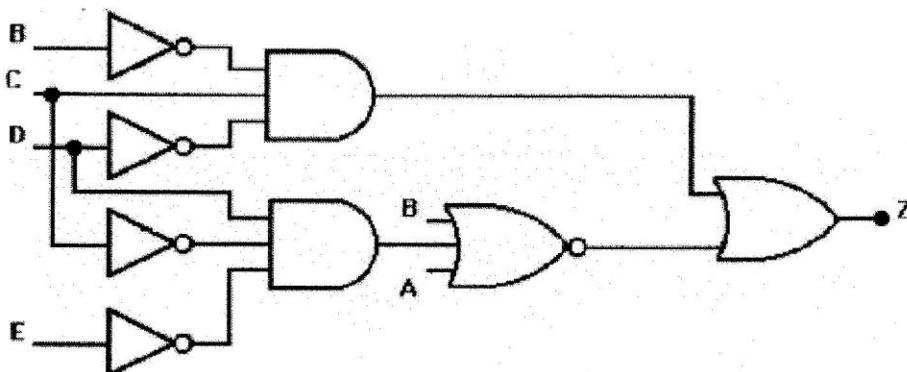
A	B	C	D	X
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



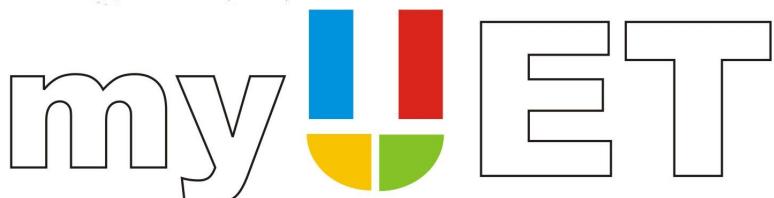
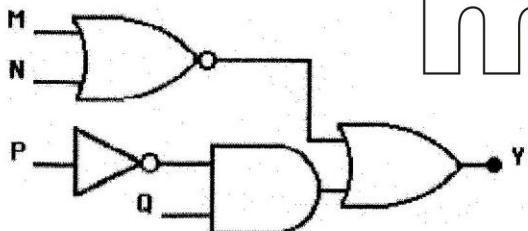
3.21 (a)



(b)



(c)



3.22 Proving theorem 15a: $X + \overline{XY} = X + Y$

$$\begin{array}{ll} X=1 & \{ 1+0\ 1=1 \} \\ Y=1 & \{ 1+0 = 1 \} \\ \{ \underline{1=1} \} & \end{array}$$

$$\begin{array}{ll} X=0 & \{ 0+1\ 0=0+0 \} \\ Y=0 & \{ 0+0 = 0 \} \\ \{ \underline{0=0} \} & \end{array}$$

$$\begin{array}{ll} X=1 & \{ 1+0\ 0=1+0 \} \\ Y=0 & \{ 1+0 = 1 \} \\ \{ \underline{1=1} \} & \end{array}$$

$$\begin{array}{ll} X=0 & \{ 0+1\ 1=0+1 \} \\ Y=1 & \{ 0+1 = 1 \} \\ \{ \underline{1=1} \} & \end{array}$$

Proving theorem 15b: $\overline{X} + XY = \overline{X} + Y$

$$\begin{array}{ll} X=1 & \{ 0+1\ 1=1 \} \\ Y=1 & \{ 0+1 = 1 \} \\ \{ \underline{1=1} \} & \end{array}$$

$$\begin{array}{ll} X=0 & \{ 1+0\ 0=1+0 \} \\ Y=0 & \{ 1+0 = 1 \} \\ \{ \underline{1=1} \} & \end{array}$$

$$\begin{array}{ll} X=1 & \{ 0+1\ 0=0+0 \} \\ Y=0 & \{ 0+0 = 0 \} \\ \{ \underline{0=0} \} & \end{array}$$

$$\begin{array}{ll} X=0 & \{ 1+0\ 1=1+0 \} \\ Y=1 & \{ 1+0 = 1 \} \\ \{ \underline{1=1} \} & \end{array}$$

3.23

- (a) $A + 1 = 1$ (b) $A \cdot A = A$ (c) $B \cdot \bar{B} = 0$ (d) $C + C = C$ (e) $X \cdot 0 = 0$ (f) $D \cdot 1 = D$
 (g) $D + 0 = D$ (h) $C + \bar{C} = 1$ (i) $G + GF = G$ (j) $Y + \bar{W}Y = Y$

3.24

(a)

$$\begin{aligned} X &= (M + N)(\bar{M} + P)(\bar{N} + \bar{P}) \\ X &= (\bar{M}\bar{M} + MP + \bar{N}\bar{M} + NP)(\bar{N} + \bar{P}) \\ X &= (MM\bar{N} + M\bar{M}\bar{P} + M\bar{P}\bar{N} + M\bar{P}\bar{P} + \bar{N}M\bar{N} + \bar{N}\bar{M}\bar{P} + N\bar{P}\bar{N} + N\bar{P}\bar{P}) \\ X &= (0 + 0 + M\bar{P}\bar{N} + 0 + 0 + N\bar{M}\bar{P} + 0 + 0) \\ X &= M\bar{P}\bar{N} + N\bar{M}\bar{P} \end{aligned}$$

(b)

$$\begin{aligned} Z &= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + B\bar{C}D \\ Z &= \bar{B}\bar{C}(\bar{A} + A + D) \\ Z &= \bar{B}\bar{C}(1 + D) \\ Z &= \bar{B}\bar{C} \end{aligned}$$

3.25

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

$$\begin{array}{ll} \text{A=1} & \left\{ \begin{array}{l} \overline{1+1} = \bar{1} \cdot 1 \\ \{ \end{array} \right. \\ \text{B=1} & \left. \begin{array}{l} \\ \end{array} \right\} \end{array}$$

$$\begin{array}{ll} \text{A=0} & \left\{ \begin{array}{l} \overline{0+0} = \bar{0} = 1 \\ \{ \end{array} \right. \\ \text{B=0} & \left. \begin{array}{l} \\ \end{array} \right\} \end{array}$$

$$\begin{array}{ll} \text{A=0} & \left\{ \begin{array}{l} \overline{0+1} = \bar{0} \cdot \bar{1} = 0 \\ \{ \end{array} \right. \\ \text{B=1} & \left. \begin{array}{l} \\ \end{array} \right\} \end{array}$$

$$\begin{array}{ll} \text{A=1} & \left\{ \begin{array}{l} \overline{1+0} = \bar{1} \cdot \bar{0} = 0 \\ \{ \end{array} \right. \\ \text{B=0} & \left. \begin{array}{l} \\ \end{array} \right\} \end{array}$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\begin{array}{ll} \text{A=1} & \left\{ \begin{array}{l} \overline{1 \cdot 1} = \bar{1} + \bar{1} = 0 \\ \{ \end{array} \right. \\ \text{B=1} & \left. \begin{array}{l} \\ \end{array} \right\} \end{array}$$

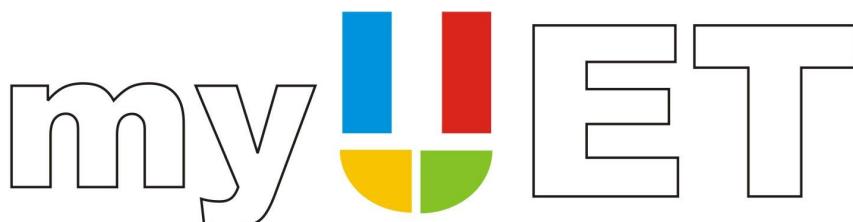
$$\begin{array}{ll} \text{A=0} & \left\{ \begin{array}{l} \overline{0 \cdot 0} = \bar{0} = 1 \\ \{ \end{array} \right. \\ \text{B=0} & \left. \begin{array}{l} \\ \end{array} \right\} \end{array}$$

$$\begin{array}{ll} \text{A=0} & \left\{ \begin{array}{l} \overline{0 \cdot 1} = \bar{0} + \bar{1} = 1 \\ \{ \end{array} \right. \\ \text{B=1} & \left. \begin{array}{l} \\ \end{array} \right\} \end{array}$$

$$\begin{array}{ll} \text{A=1} & \left\{ \begin{array}{l} \overline{1 \cdot 0} = \bar{1} + \bar{0} = 1 \\ \{ \end{array} \right. \\ \text{B=0} & \left. \begin{array}{l} \\ \end{array} \right\} \end{array}$$

3.26

- (a) $\overline{\overline{ABC}} = \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} = A + \bar{B} + \bar{C}$
 (b) $\overline{\overline{A + BC}} = \overline{\overline{A}}(\overline{\overline{B}} + \overline{\overline{C}}) = A(B + \bar{C})$
 (c) $\overline{\overline{ABCD}} = \overline{\overline{AB}} + \overline{\overline{CD}} = \overline{\overline{A}} + \overline{\overline{B}} + CD$
 (d) $\overline{\overline{A + B}} = \overline{\overline{AB}} = \overline{AB}$
 (e) $\overline{\overline{AB}} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$
 (f) $\overline{\overline{A + \overline{C} + \overline{D}}} = \overline{\overline{ACD}} = ACD$



$$(g) \overline{A(\overline{B+C})D} = \overline{A} + \overline{\overline{B+C}} + \overline{D} = \overline{A} + B + \overline{C} + \overline{D}$$

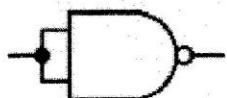
$$(h) \overline{(M+N)(\overline{M}+N)} = \overline{MN} + \overline{M}\overline{N}$$

$$(i) \overline{\overline{ABCD}} = \overline{ABC} + \overline{D} = (\overline{A} + \overline{B})C + \overline{D}$$

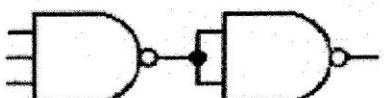
3.27

$$X = \overline{(A+B)\overline{BC}} = \overline{A} + \overline{B} + \overline{\overline{BC}} = A + B + B + \overline{C} = A + B + \overline{C}$$

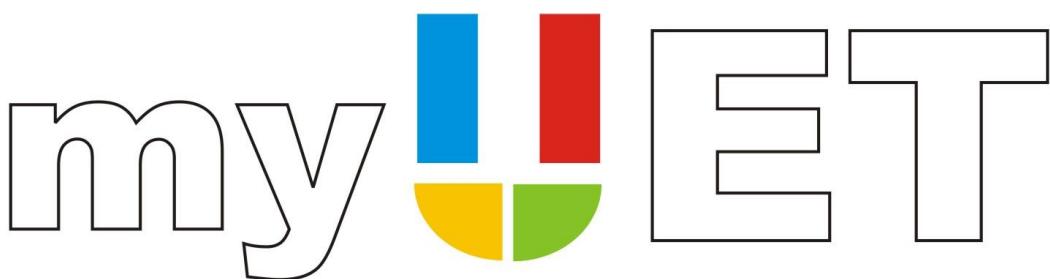
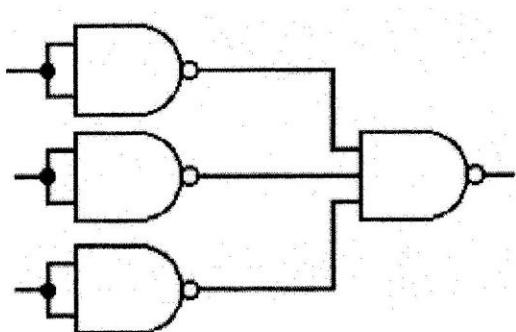
3.28 Change each inverter to:



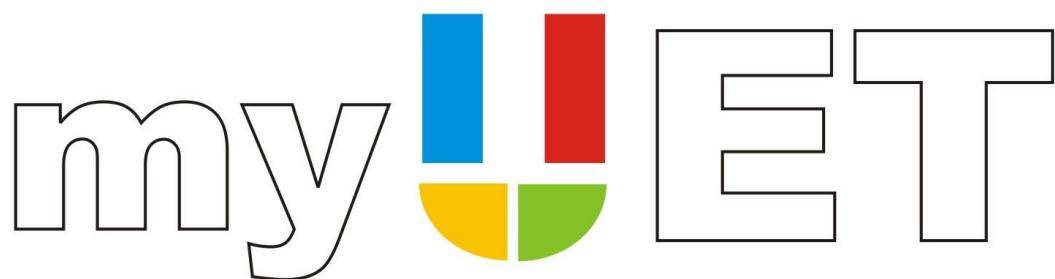
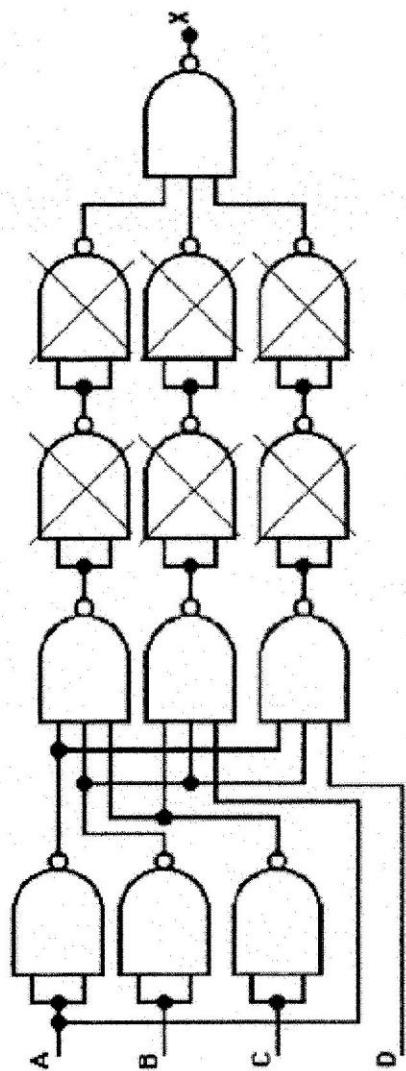
Change each AND to:



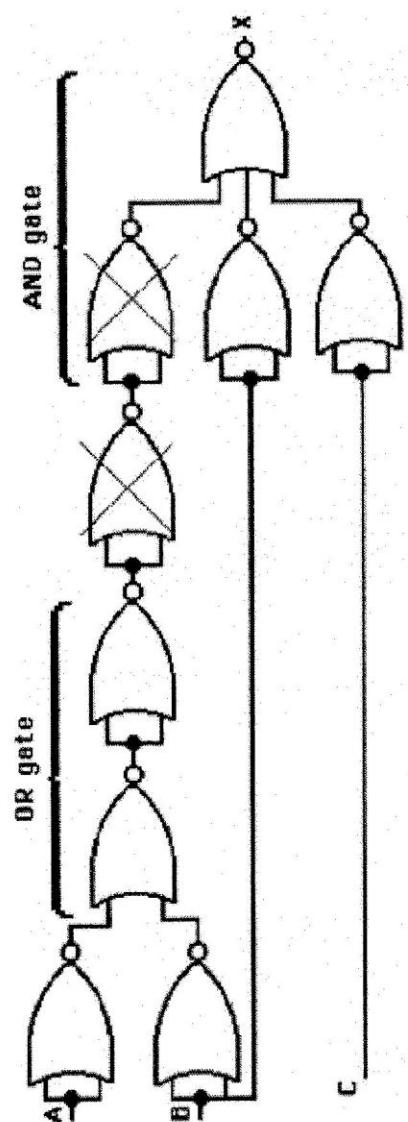
Change each OR to:



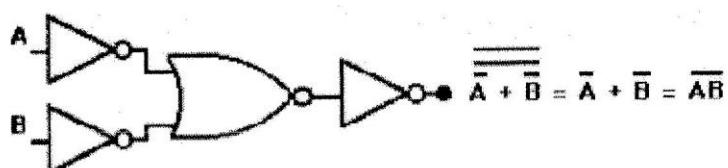
By canceling double INVERTERS, the result is: $X = \overline{ABC} + A\overline{BC} + \overline{ABD}$



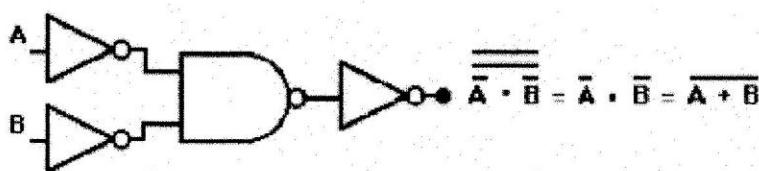
3.29 X=ABC



3.30

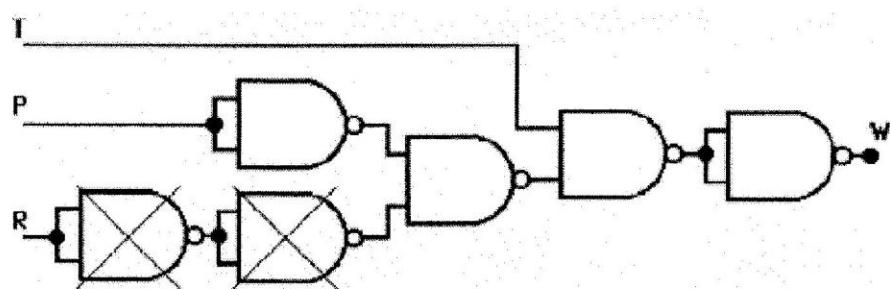


3.31



- 3.32 (a) The warning light W will be activated, when temperature (T) is $>200^{\circ}\text{F}$ **and** either the pressure (P) is >220 p.s.i., **or** the speed (R) is < 4800 r.p.m. In conclusion, **W=1 when T=1 and either P=1 or R=0.**

(b)

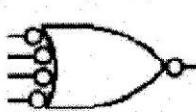


3.33

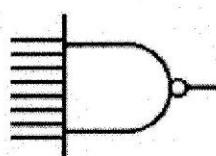
(a) NOR gate



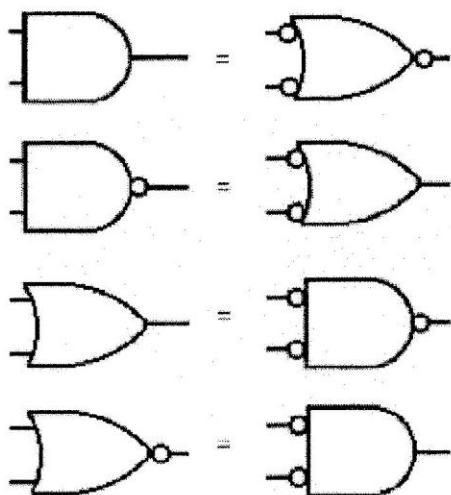
(b) AND gate



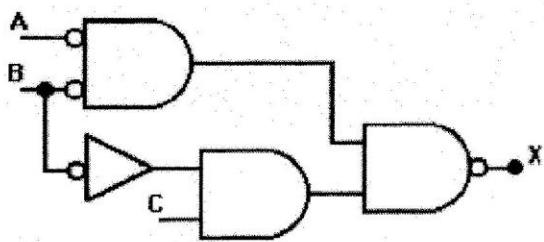
(c) NAND gate



3.34



3.35 (a)

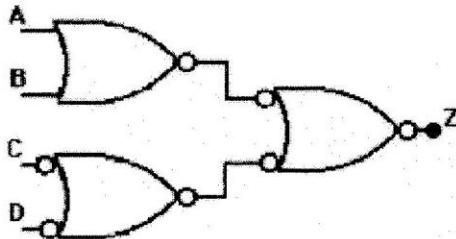


(b)

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

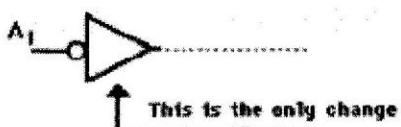
3.36 (a) Z is HIGH only when A=B=0 and C=D=1.

(b)



Z will be LOW when A or B is HIGH, or when C or D is LOW.

3.37



3.38 X will go HIGH when E=1, or D=0, or C=B=0, or when B=1 and A=0.

3.39 (a) X is asserted (active) HIGH.
 (b) Z is asserted (active) LOW.



3.40

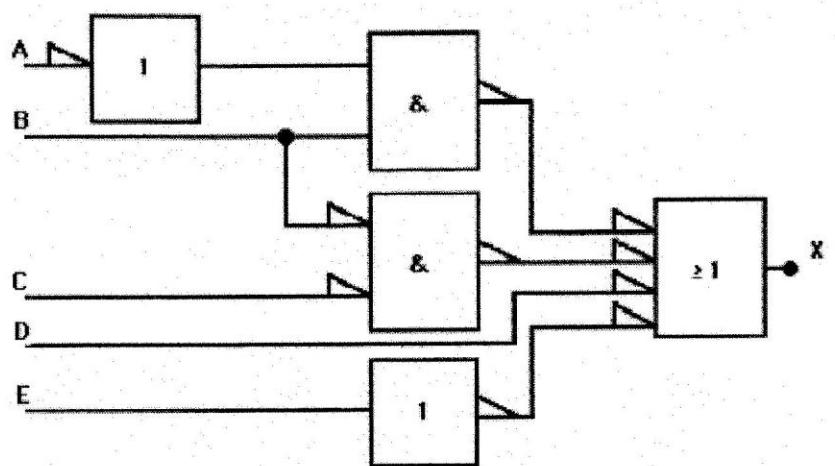
E	D	C	B	A	X
0	0	0	0	0	1
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

3.41 LIGHT = LOW when A=B=1, or when A=B=0.

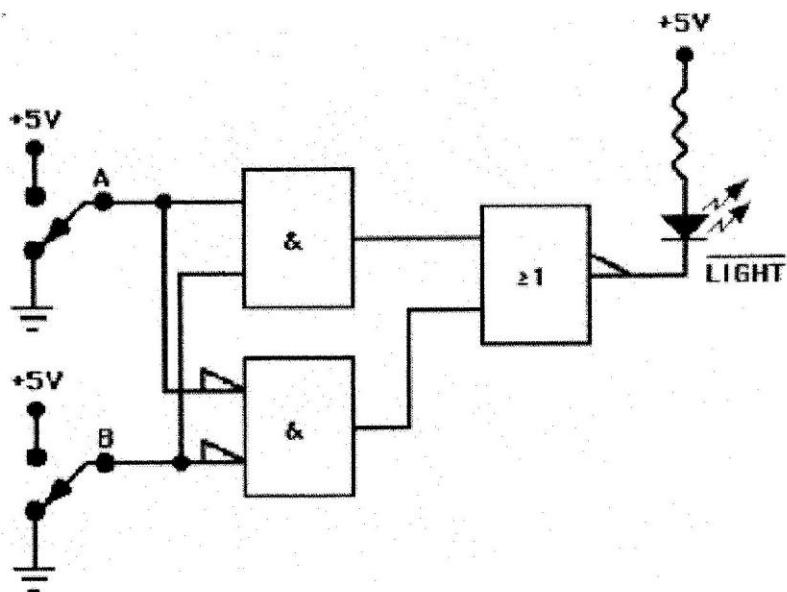
B	A	<u>LIGHT</u>
0	0	0
0	1	1
1	0	1
1	1	0



3.42 (a)



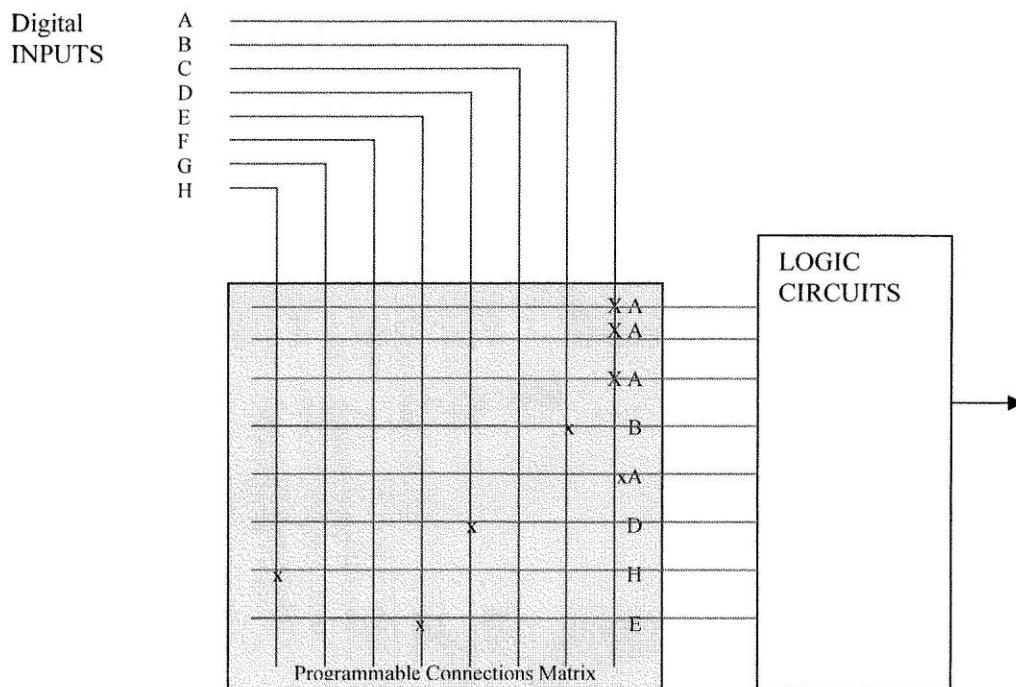
(b)



3.43

- (a) FALSE (b) TRUE (c) FALSE (d) TRUE (e) FALSE (f) FALSE (g) TRUE
- (h) FALSE (i) TRUE (j) TRUE

3.44



3.45

AHDL

SUBDESIGN prob3_45

(

a,b,c	:INPUT;	--define inputs to block
x1,y,z	:OUTPUT;	--define block output

)

```

BEGIN
x1 = a # b;
y = !(a & b);
z = a # b # c;
END;
```

VHDL

ENTITY prob3_45 IS

PORT (a, b, c	:IN bit;	--define inputs to block
x, y, z	:OUT bit);	--define block output

END prob3_45 ;

ARCHITECTURE ckt OF prob3_45 IS

```

BEGIN
x<= a OR b;
y <= NOT(a AND b);
z <= a OR b OR c;
END ckt;
```

-- logic descriptions



3.46

(a) AHDL

```
SUBDESIGN prob3_46
(
    rd, rom_a, rom_b, ram           :INPUT;      --define inputs to block
    mem                           :OUTPUT;     --define block output
)
BEGIN
    mem = !rd & (!rom_a # !rom_b) # !ram;
END;
```

(a) VHDL

```
ENTITY prob3_46 IS
    PORT (rd, rom_a, rom_b, ram           :IN bit;      --define inputs to block
          mem                         :OUT bit);   --define block output
END prob3_46;

ARCHITECTURE ckt OF prob3_46 IS
BEGIN
    mem <= (NOT rd) AND ((NOT rom_a) OR (NOT rom_b) OR (NOT ram));
END ckt;
```

(b) AHDL

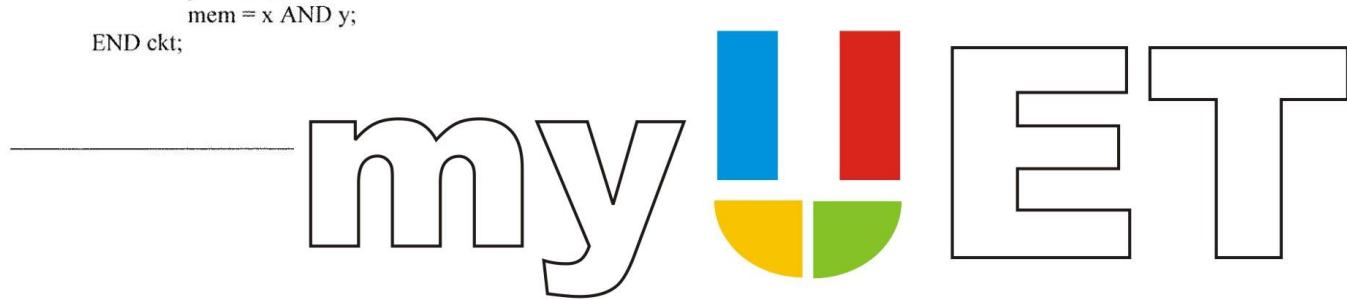
```
SUBDESIGN prob3_46
(
    rd, rom_a, rom_b, ram           :INPUT;      --define inputs to block
    mem                           :OUTPUT;     --define block output
)
VARIABLE
    v,w,x1,y :NODE;
BEGIN
    x1 = !rd;
    w = !rom_a # !rom_b;
    v = !ram;
    y = w # v;
    mem = x1 & y;
END;
```

(b) VHDL

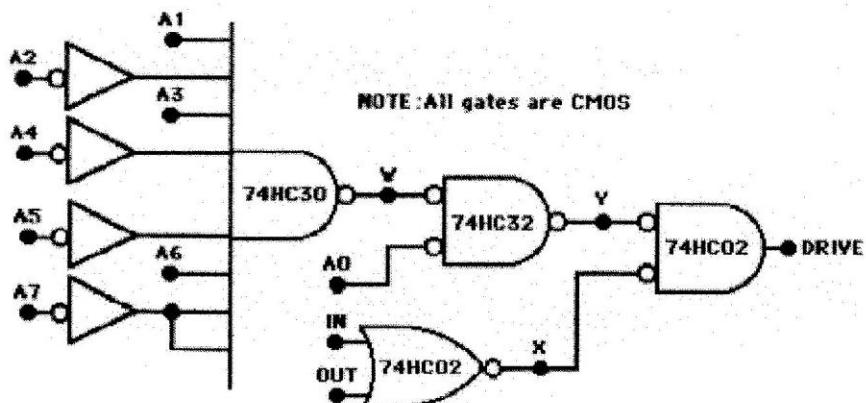
```
ENTITY prob3_46 IS
    PORT (rd, rom_a, rom_b, ram           :IN bit;      --define inputs to block
          mem                         :OUT bit);   --define block output
END prob3_46;

ARCHITECTURE ckt OF prob3_46 IS
SIGNAL v,w,x,y :BIT;

BEGIN
    x <= NOT rd;
    w <= (NOT rom_a) OR (NOT rom_b);
    v <= NOT ram;
    y <= w OR v;
    mem = x AND y;
END ckt;
```



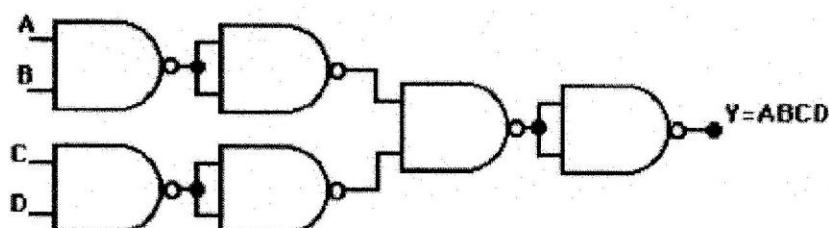
3.47



3.48



3.49



CHAPTER FOUR - Combinational Logic Circuits

4.1

(a) $x = ABC + \overline{AC} = C(AB + \overline{A}) = C(\overline{A} + B)$

(b) $y = (Q + R)(\overline{Q} + \overline{R}) = Q\overline{Q} + Q\overline{R} + R\overline{Q} + R\overline{R} = Q\overline{R} + R\overline{Q}$

(c) $w = ABC + A\overline{B}\overline{C} + \overline{A} = AC(B + \overline{B}) + \overline{A} = AC + \overline{A} = \overline{A} + C$

(d)

$$q = \overline{RST(R + S + T)}$$

$$q = (\overline{R} + \overline{S} + \overline{T})(RST)$$

$$q = \overline{RRST} + \overline{SRST} + \overline{TRST}$$

$$q = \overline{RST} + \overline{RST} + \overline{RST}$$

$$q = \overline{RST}$$

(e)

$$x = \overline{ABC} + \overline{ABC} + ABC + \overline{ABC} + A\overline{BC}$$

$$x = \overline{ABC} + BC(A + \overline{A}) + A\overline{B}(C + \overline{C})$$

$$x = \overline{ABC} + BC + A\overline{B}$$

$$x = BC + \overline{B}(\overline{AC} + A) = BC + \overline{B}(A + \overline{C})$$

One possibility:

(f)

$$z = (B + \overline{C})(\overline{B} + C) + \overline{\overline{A} + B + \overline{C}}$$

$$z = \overline{BB} + BC + \overline{BC} + \overline{CC} + \overline{\overline{ABC}} =$$

$$z = BC + \overline{BC} + A\overline{BC}$$

$$z = BC + \overline{B}(\overline{C} + AC)$$

$$z = BC + \overline{B}(\overline{C} + A)$$

$$z = BC + \overline{BC} + A\overline{B}$$

(g)

$$y = \overline{(C + D)} + \overline{ACD} + A\overline{BC} + \overline{ABCD} + ACD$$

$$y = \overline{CD} + \overline{ACD} + \overline{ABC} + \overline{ABCD} + ACD$$

$$y = \overline{CD} + \overline{CD}(\overline{A} + A) + \overline{ABC} + \overline{ABCD}$$

$$y = \overline{CD} + \overline{CD} + A\overline{BC} + \overline{ABCD}$$

$$y = \overline{D}(\overline{C} + C) + A\overline{BC} + \overline{ABCD}$$

$$y = \overline{D} + A\overline{BC} + \overline{ABCD}$$

$$y = \overline{D} + A\overline{BC} + \overline{ABC}$$



(h)

$$\begin{aligned}x &= AB(\overline{CD}) + \overline{ABD} + \overline{BCD} \\x &= AB(C + \overline{D}) + \overline{ABD} + \overline{BCD} \\x &= ABC + ABD + \overline{ABD} + \overline{BCD}\end{aligned}$$

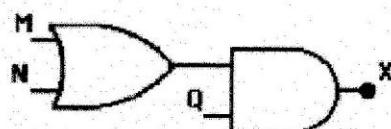
4.2

$$X = \overline{\overline{MNQ} \cdot \overline{\overline{MNQ}} \cdot \overline{\overline{MNQ}}}$$

$$X = MNQ + M\overline{N}Q + \overline{M}NQ$$

$$X = MQ + NQ$$

$$X = Q(M + N)$$



4.3

$$X = \overline{(M + N + Q)} + \overline{(M + \overline{N} + Q)} + \overline{(\overline{M} + N + Q)}$$

$$X = (M + N + Q)(M + \overline{N} + Q)(\overline{M} + N + Q)$$

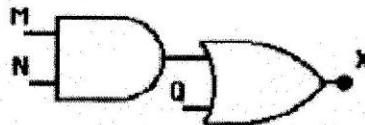
$$X = (MM + M\overline{N} + MQ + NM + N\overline{N} + NQ + QM + Q\overline{N} + QQ)(\overline{M} + N + Q)$$

$$X = (M + M\overline{N} + MQ + NM + NQ + QM + Q\overline{N} + Q)(\overline{M} + N + Q)$$

$$X = (M + Q)(\overline{M} + N + Q)$$

$$X = M\overline{M} + MN + MQ + \overline{M}Q + QN + QQ$$

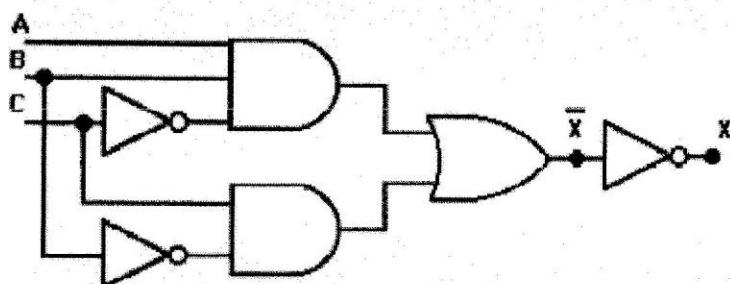
$$X = MN + Q$$



4.4 Use \overline{X} since this would give only three terms.

$$\overline{X} = \overline{ABC} + \overline{A\overline{B}C} + A\overline{B}\overline{C}$$

$$\overline{X} = \overline{BC} + ABC$$

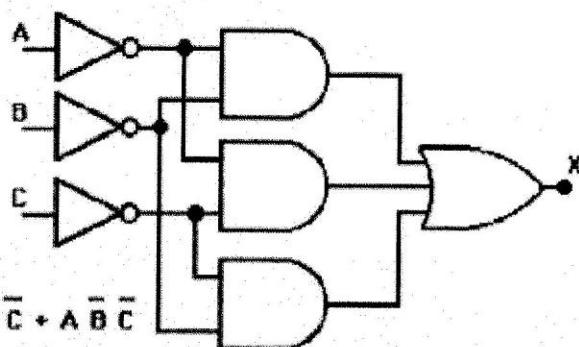


Alternate solution using S-of-P expression for X would be: $X = \overline{AB} + \overline{BC} + BC$

4.5

A	B	C	X
0	0	0	1 ($\bar{A} \bar{B} \bar{C}$)
0	0	1	1 ($\bar{A} B \bar{C}$)
0	1	0	1 ($\bar{A} \bar{B} C$)
0	1	1	0
1	0	0	1 ($A \bar{B} \bar{C}$)
1	0	1	0
1	1	0	0
1	1	1	0

$$X = \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + \bar{A} \bar{B} C + A \bar{B} \bar{C}$$



By adding the term $\bar{A} \bar{B} \bar{C}$ three times and then factoring, the following is obtained:

$$X = \bar{A} \bar{B} (C + \bar{C}) + \bar{A} C (B + \bar{B}) + \bar{B} C (A + \bar{A})$$

$$X = \bar{A} \bar{B} + \bar{A} C + \bar{B} C$$

4.6 Make the following assumptions:

A - It's 5:00 or later; **B** - All machines are shut down; **C** - It's Friday

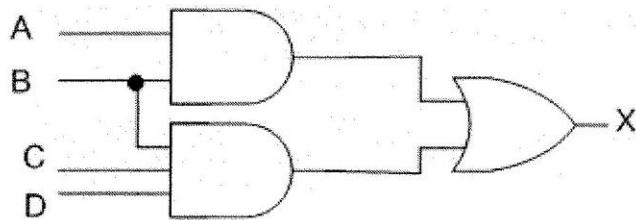
D - Production run for the day is complete

Output Y assumes all variables that are not mentioned in the conditions of the story problem must be zero to blow the horn. Output X assumes that all variables that are not mentioned in the conditions of the story problem can be either 1 or 0 in order to blow the horn.

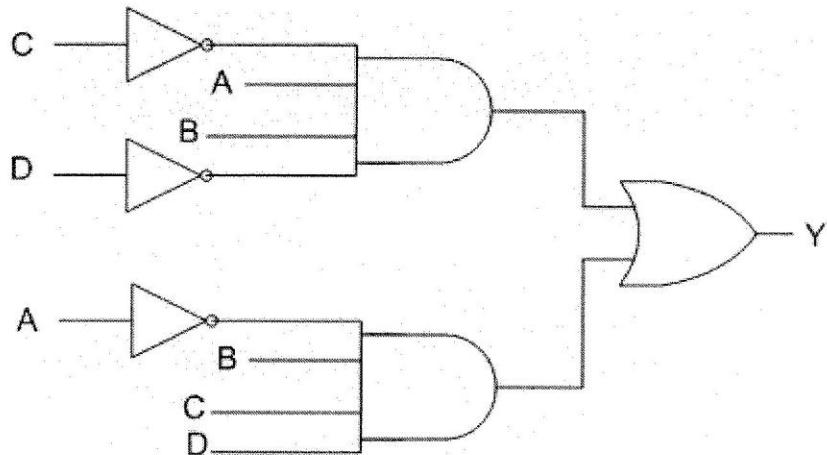
D	C	B	A	X	Y
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	1	1
1	1	1	1	1	0

$$X = ABC\bar{D} + ABC\bar{D} + ABCD + \bar{A}BCD + ABCD$$

$$X = AB + BCD$$



$$Y = \overline{ABC}\overline{D} + \overline{AB}\overline{CD}$$



4.7

<i>A₃</i>	<i>A₂</i>	<i>A₁</i>	<i>A₀</i>	<i>X</i>
0	0	0	0	0
0	0	0	1	0
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
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0
1	1	1	1	0

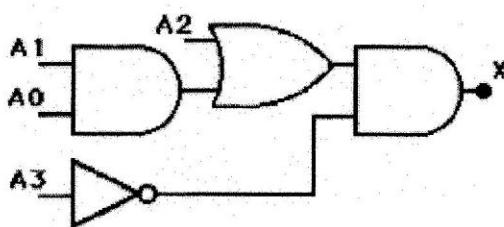
By inspection, *X* will be 1 whenever *A₃*=0, *A₂*=1; or when *A₃*=*A₂*=0, while *A₁*=*A₀*=1.
Thus, we can write:

$$X = \overline{A_3} A_2 + \overline{A_3} \overline{A_2} A_1 A_0$$

$$X = \overline{A_3}(A_2 + \overline{A_2} A_1 A_0)$$

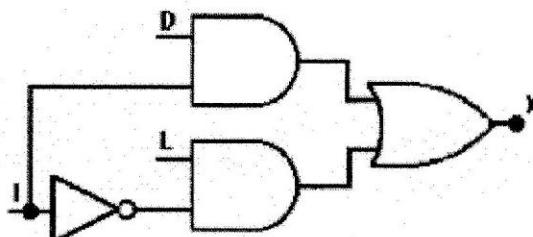
$$X = \overline{A_3}(A_2 + A_1 A_0)$$

The same result can be obtained by writing the S-of-P expression and then simplifying it.



4.8 Door = D; Ignition = I; Lights = L

L	I	D	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

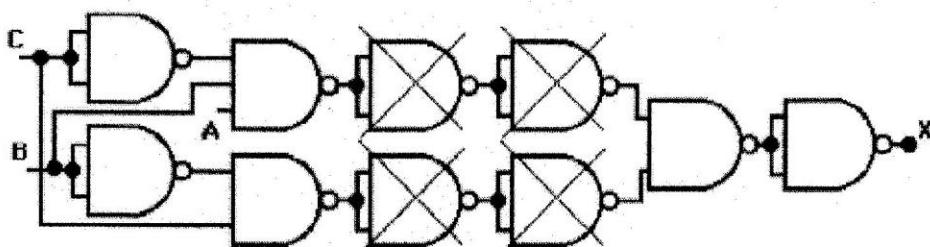


$$X = \overline{L}ID + L\overline{I}D + L\overline{I}D + LID$$

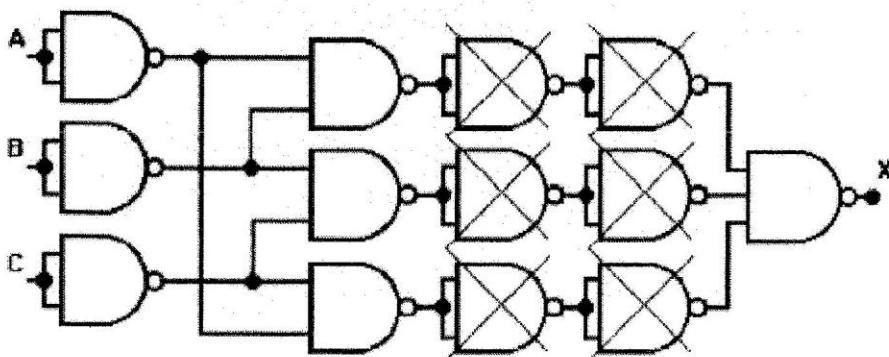
$$X = ID(\overline{L} + L) + LI(\overline{D} + D)$$

$$X = ID + LI$$

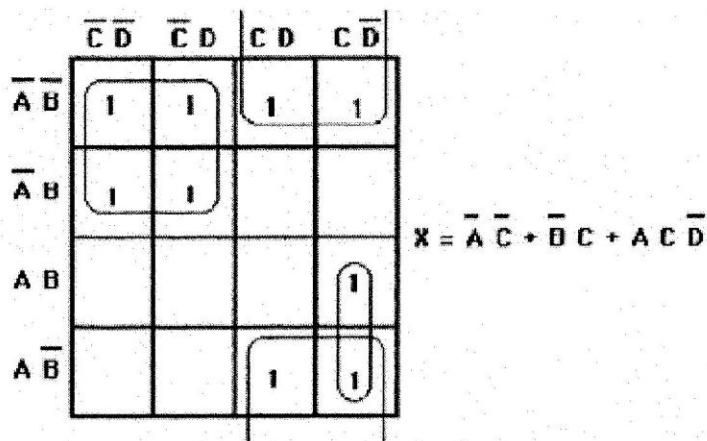
4.9 Change each gate to its NAND equivalent and then cancel double inversions.



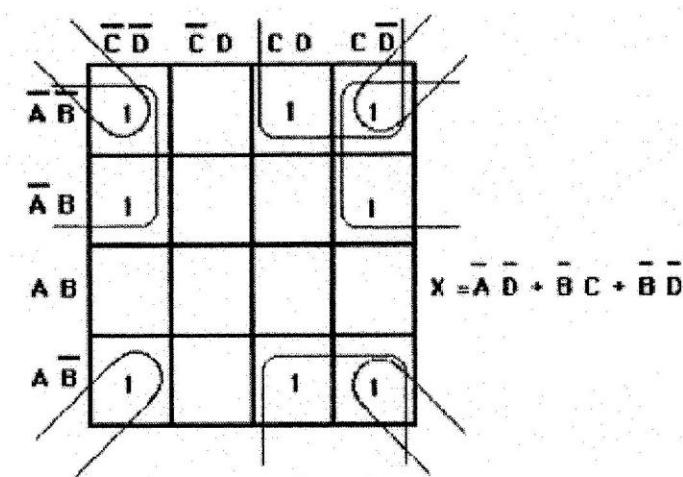
4.10 Change each gate to its NAND equivalent and then cancel double inversions.



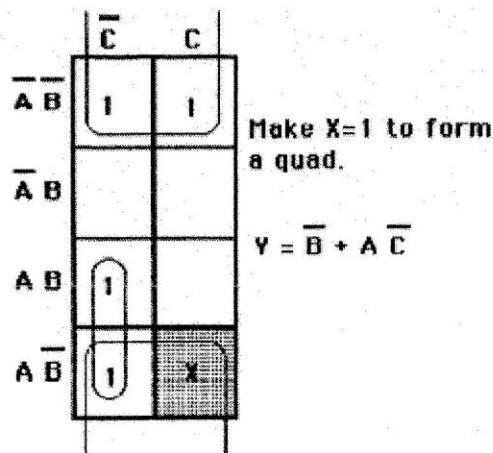
4.11 (a)



(b)



(c)



4.12 $Y = \overline{A}$

\overline{A}	\overline{B}	B
A	1	1
A	0	0

4.13 $X = \overline{BC} + BC + \overline{AB}$ Other solution: $X = \overline{BC} + BC + \overline{AC}$

\overline{A}	\overline{B}	\overline{C}	B	C	B	C	B	\overline{C}
A	1			1		1		
A	1				1			

4.14 (a) $X = \overline{ABC} + \overline{ABC} + ABC + A\overline{BC} + A\overline{BC}$

\overline{A}	\overline{B}	\overline{C}	B	C	B	C	B	\overline{C}
A	1				1			
A	1		1			1		

$X = \overline{BC} + BC + \overline{AB}$ Other solution: $X = \overline{BC} + BC + AC$

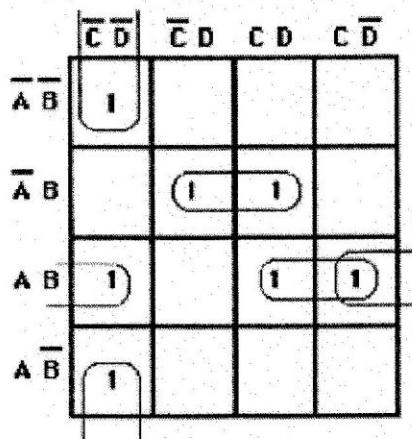
(b) $Y = \overline{CD} + \overline{ACD} + A\overline{BC} + A\overline{BCD} + ACD$

\overline{A}	\overline{B}	\overline{C}	D	\overline{C}	D	C	\overline{D}
A	B	1			1		
A	B	1				1	
A	B	1					1
A	B		1				

$Y = \overline{D} + A\overline{BC} + A\overline{BC}$

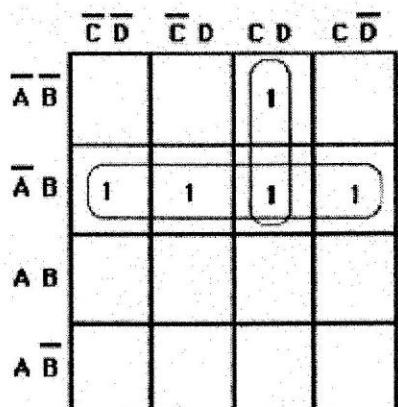
(c) One possibility:

$$X = A B C + A B \bar{D} + \bar{A} B D + \bar{B} \bar{C} \bar{D}$$



$$X = \bar{A} B D + A B C + A B \bar{D} + C D \bar{B}$$

4.15 For visual convenience, let $A_3=A$, $A_2=B$, $A_1=C$, $A_0=D$

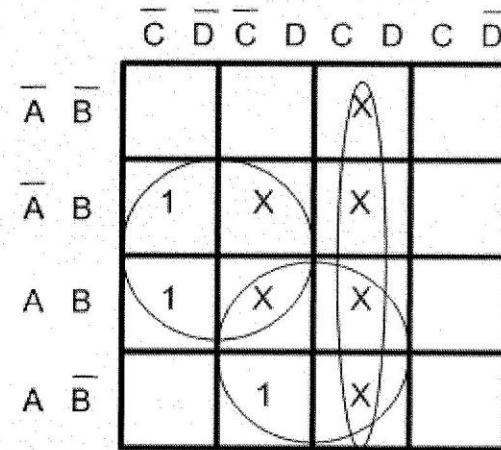


$$X = \bar{A} \bar{B} + \bar{A} C \bar{D}$$

$$X = \bar{A}_3 \cdot A_2 + \bar{A}_3 \cdot A_1 \cdot A_0$$

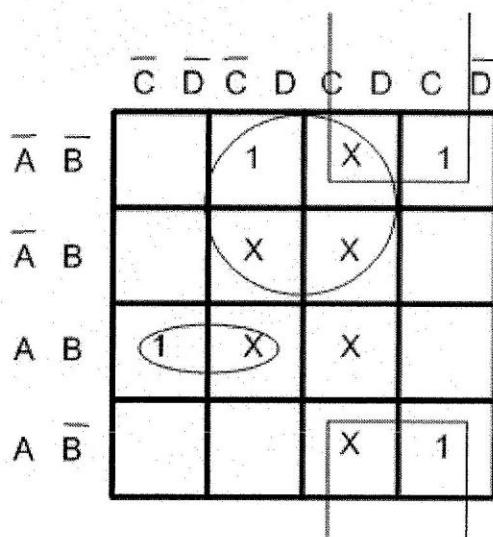
4.16 (a) $X = \overline{BC} + AD$

D	C	B	A	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	x
1	0	1	1	x
1	1	0	0	x
1	1	0	1	x
1	1	1	0	x
1	1	1	1	x



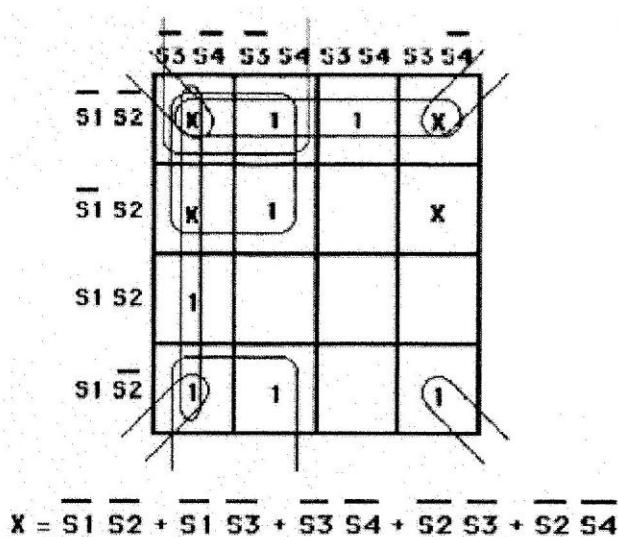
(b) $X = \overline{BC} + \overline{AD} + ABC\overline{C}$

D	C	B	A	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	x
1	0	1	1	x
1	1	0	0	x
1	1	0	1	x
1	1	1	0	x
1	1	1	1	x

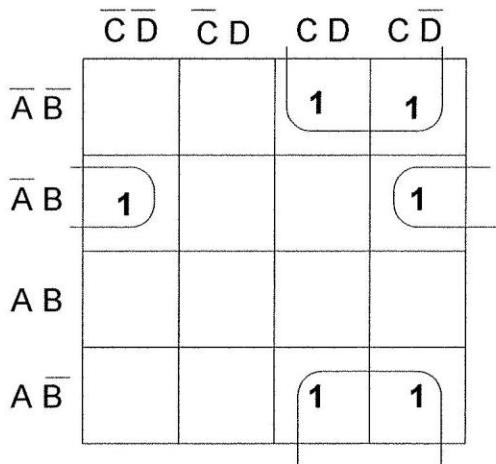


4.17

S4	S3	S2	S1	X
0	0	0	0	X
0	0	0	1	$\overline{S1} \ S2 \ \overline{S3} \ \overline{S4}$
0	0	1	0	X
0	0	1	1	$\overline{S1} \ S2 \ \overline{S3} \ \overline{S4}$
0	1	0	0	X
0	1	0	1	$\overline{S1} \ S2 \ S3 \ \overline{S4}$
0	1	1	0	X
0	1	1	1	0
1	0	0	0	$\overline{S1} \ S2 \ \overline{S3} \ \overline{S4}$
1	0	0	1	$\overline{S1} \ S2 \ \overline{S3} \ S4$
1	0	1	0	$\overline{S1} \ S2 \ S3 \ \overline{S4}$
1	0	1	1	0
1	1	0	0	$\overline{S1} \ S2 \ S3 \ S4$
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



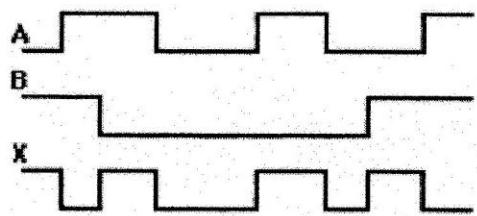
4.18 $z = \overline{A}\overline{B}\overline{D} + \overline{B}C$



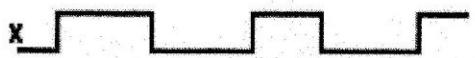
4.19 In Example 4.3 of your textbook, after the DeMorgan part is completed, we have:

$$\begin{aligned}
 z &= \overline{ABC} + \overline{ACD} + \overline{ABCD} + ABC \\
 z &= \overline{ABC} + \overline{ACD}(B + \overline{B}) + \overline{ABCD} + ABC \\
 z &= \overline{ABC} + \overline{ABCD} + \overline{ACD} + \overline{ABCD} + ABC \\
 z &= \overline{ABC} + \overline{ABCD} + \overline{ACD} + \overline{ABCD} + ABC \\
 z &= \overline{BC}(\overline{A} + A + \overline{AD}) + \overline{ABD}(C + \overline{C}) \\
 z &= \overline{BC} + \overline{ABD}
 \end{aligned}$$

- 4.20** (a) Output X will be HIGH only when A and B are at different levels.



- (b) With B held LOW, X=A.



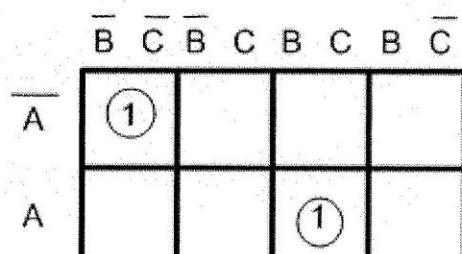
- (c) With B held HIGH, X = \overline{A} .



- 4.21** X will be HIGH when A \neq B, B=C, and C=1. Thus, C=1, B=1, A=0 is the only input condition that produces X=1.

4.22 (a) $X = ABC + \overline{ABC}$

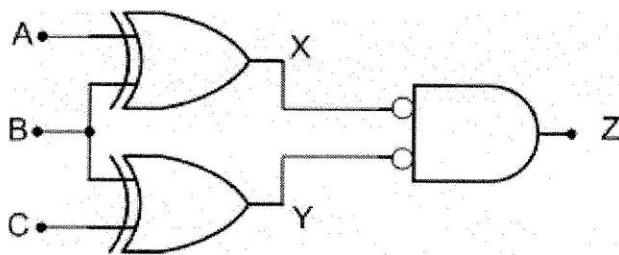
C	B	A	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



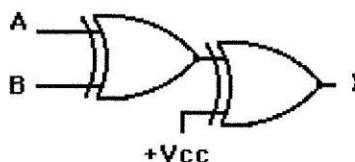
- (b)** To find if A=B=C:

1. $X = A \oplus B$ (X is Low when A=B)
2. $Y = B \oplus C$ (Y is Low when B=C)

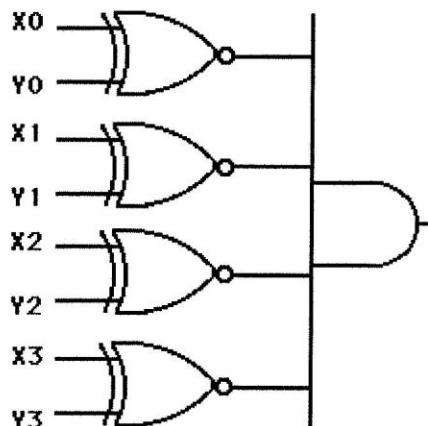
A=B=C when both 1 & 2 are true.



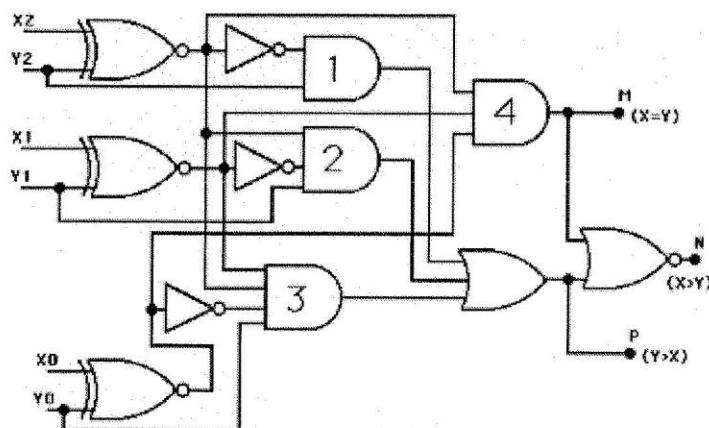
4.23



4.24



- 4.25 One possibility is on the next page. Note the use of the XNOR gates and AND gate 4 to determine when the two numbers are equal; that is, when $X_2=Y_2$, $X_1=Y_1$ and $X_0=Y_0$ simultaneously. AND gates 1,2,3 and the OR gate are used to sense when $Y_2 \ Y_1 \ Y_0 > X_2 \ X_1 \ X_0$. The NOR gate simply uses the fact that if neither M nor P is HIGH then it must be true that $X_2 \ X_1 \ X_0 > Y_2 \ Y_1 \ Y_0$, and therefore N=1.



4.26

INPUTS				OUTPUTS			
Y1	Y0	X1	X0	Z3	Z2	Z1	Z0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	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	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Four outputs are required since the largest product will be 1001 (nine) for the case where $Y_1 Y_0 = 11$ (three), and $X_1 X_0 = 11$ (three). Z0 is the LSB of the output.

Since there are four separate outputs, then four separate circuits must be designed; one for each output.

Output Z3: $Z_3 = 1$ only for single case in the T-T. Thus, $Z_3 = Y_1 Y_0 X_1 X_0$

Output Z2: Z2 is HIGH for three cases. Thus,

$$Z_2 = Y_1 \bar{Y}_0 X_1 \bar{X}_0 + Y_1 Y_0 \bar{X}_1 \bar{X}_0 + Y_1 \bar{Y}_0 \bar{X}_1 X_0$$

$$Z_2 = Y_1 X_1 (\bar{Y}_0 + \bar{X}_0) = Y_1 X_1 (\bar{Y}_0 \bar{X}_0)$$

Output Z1: Z1 is HIGH for six cases. Thus,

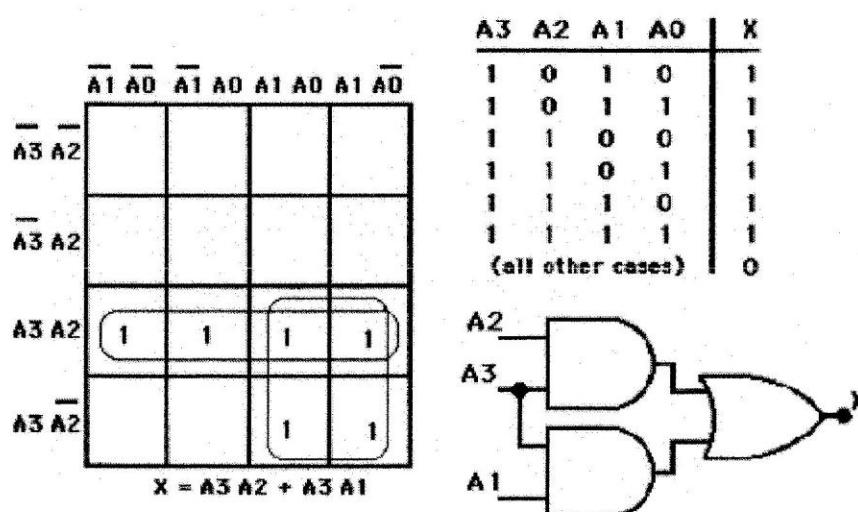
$$Z_1 = \bar{Y}_1 Y_0 X_1 \bar{X}_0 + \bar{Y}_1 \bar{Y}_0 X_1 X_0 + Y_1 \bar{Y}_0 \bar{X}_1 X_0 + Y_1 \bar{Y}_0 X_1 \bar{X}_0 + Y_1 Y_0 \bar{X}_1 \bar{X}_0 + Y_1 Y_0 \bar{X}_1 X_0$$

$$Z_1 = Y_0 X_1 (Y_1 + X_0) + Y_1 X_0 (\bar{Y}_0 + X_1)$$

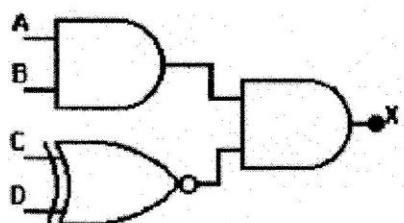
Output Z0: Z0 is HIGH for four cases.

$$Z_0 = \bar{Y}_1 \bar{Y}_0 \bar{X}_1 X_0 + \bar{Y}_1 Y_0 \bar{X}_1 X_0 + Y_1 \bar{Y}_0 \bar{X}_1 X_0 + Y_1 Y_0 \bar{X}_1 X_0 . \text{ Thus, } Z_0 = Y_0 X_0$$

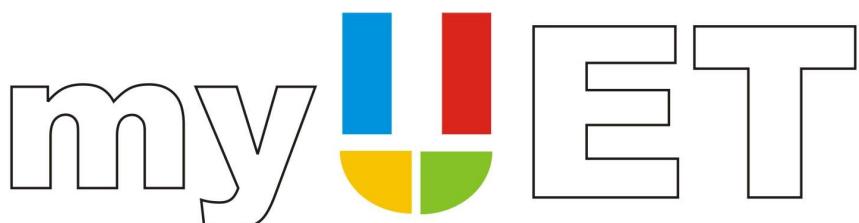
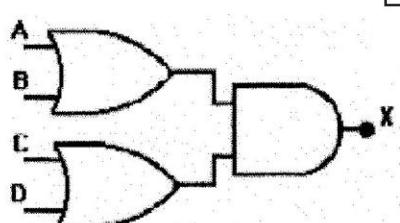
4.27



4.28



4.29



4.30

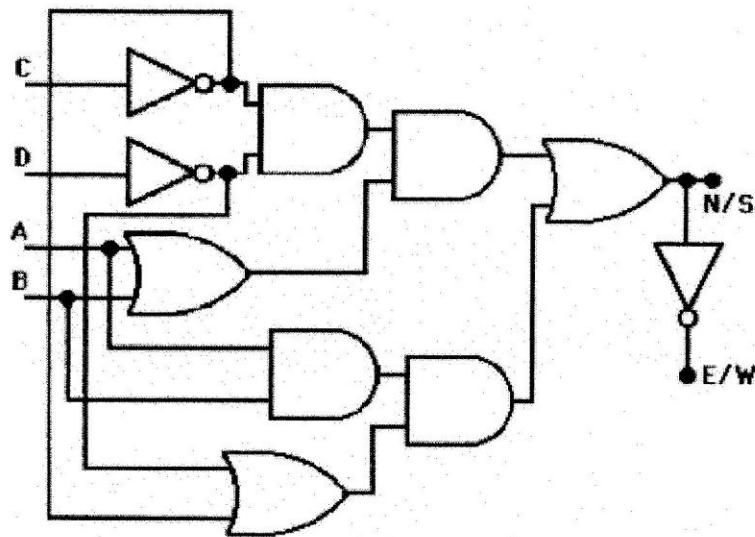
D	C	B	A	E/W	N/S
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	\overline{ABCD}
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	$\overline{ABC}\overline{D}$
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	$\overline{ABC}\overline{D}$
1	1	0	1	0	$\overline{ABC}\overline{D}$
1	1	1	0	0	$\overline{ABC}\overline{D}$
1	1	1	1	1	0

Since there are only five cases when N/S=1, we will design for N/S.

$$N/S = \overline{ABCD} + A\overline{BCD} + AB\overline{CD} + ABC\overline{D} + ABCD$$

This can be simplified to: $N/S = \overline{CD}(A+B) + AB(\overline{C}+\overline{D})$

Obviously, E / W = $\overline{N/S}$

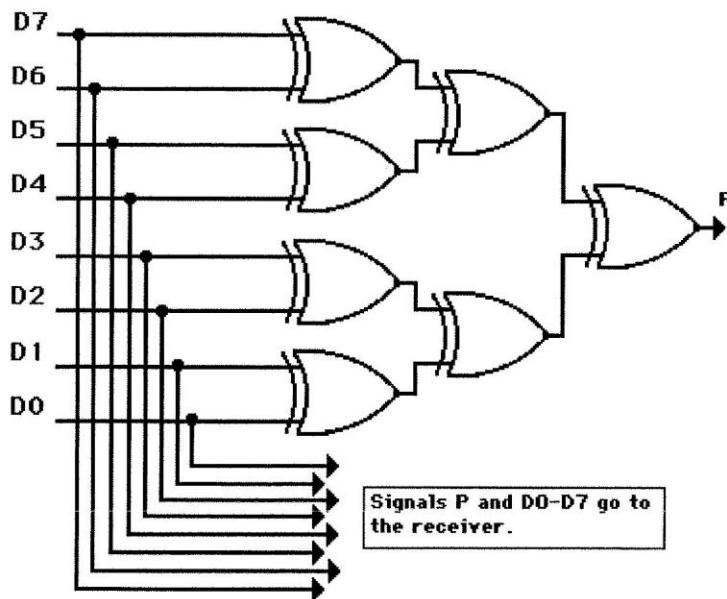


4.31 (a) Parity Generator: To modify the circuit of figure 4-25 (a) to an "Odd Parity Generator" all that is needed is an inverter at the output.

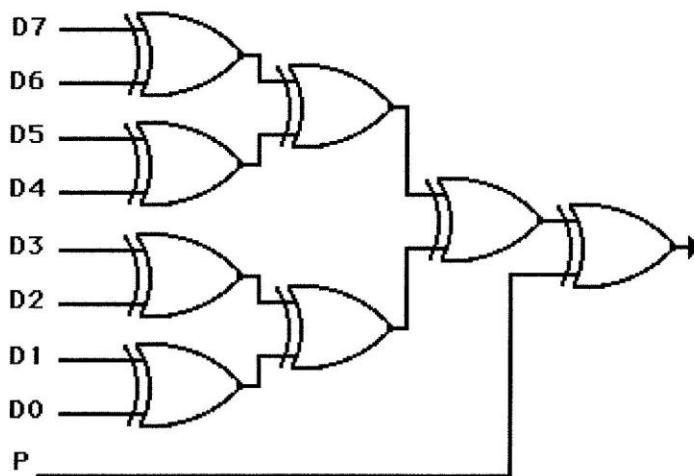
Odd Parity Checker: To modify the circuit of figure 4-25 (b) to an "Odd Parity Checker" the 2-input exclusive-OR gates should be changed to 2-input exclusive-NOR gates.

(b)

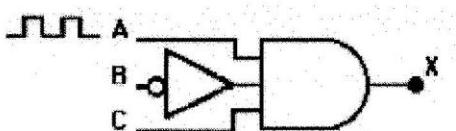
Even Parity Generator



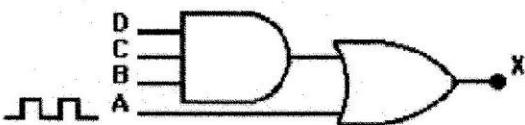
Even Parity Checker



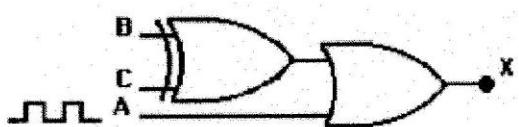
- 4.32** (a) When all of the other inputs to the OR gate are in the LOW state the logic signal will pass through to its output unchanged.
- (b) When all of the other inputs to the AND gate are in the HIGH state the logic signal will pass through to its output unchanged.
- (c) When all of the other inputs to the NAND gate are in the HIGH state the logic signal will pass through to its output INVERTED.
- (d) When all of the other inputs to the NOR gate are in the LOW state the logic signal will pass through to its output INVERTED.
- 4.33** (a) No. A logic circuit must have two inputs in order to be used as an enable/disable circuit.
 (b) No. The control input of an XOR gate can be either HIGH or LOW. If the control input is LOW the signal at the other input reaches the gate's output unaffected. If the control input is HIGH the signal at the other input reaches the gate's output INVERTED.
- 4.34** Use an AND gate that is enabled when B=0, C=1. X=A only if B=0, C=1



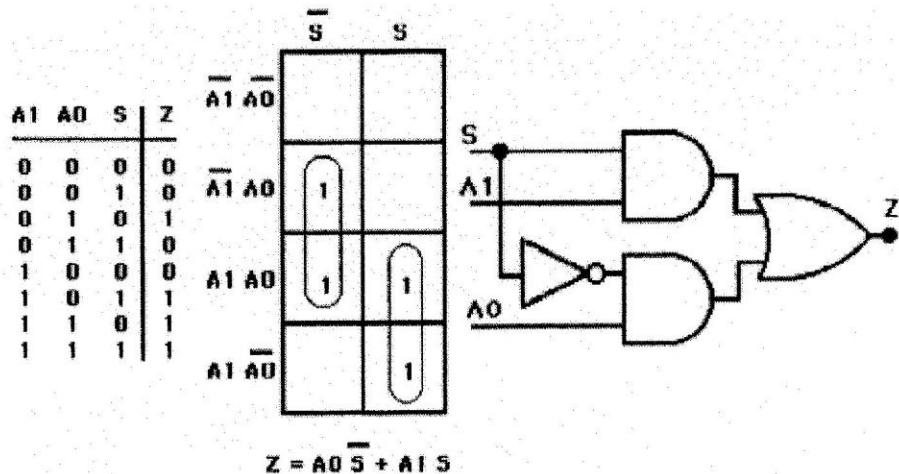
- 4.35** Use an OR gate since output is to be HIGH when inhibited. X=A only if BCD ≠ 1. X=1 when BCD = 1



4.36 X=A when B=C. X=1 when B≠C



4.37



4.38

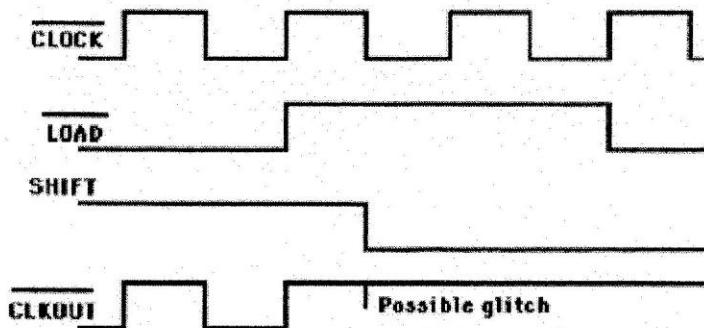
		$\overline{Y_1}$	$\overline{Y_0}$	$\overline{Y_1}$	Y_0	Y_1	Y_0	Y_1	Y_0
		$\overline{X_1}$	X_0	1					
		$\overline{X_1}$	X_0		1				
		X_1	X_0			1			
		X_1	$\overline{X_0}$				1		

No pairs,
no quads,
no octets.

$$Z = \overline{X_1} \overline{X_0} \overline{Y_1} Y_0 + \overline{X_1} X_0 \overline{Y_1} Y_0 + X_1 X_0 Y_1 Y_0 + X_1 \overline{X_0} Y_1 \overline{Y_0}$$

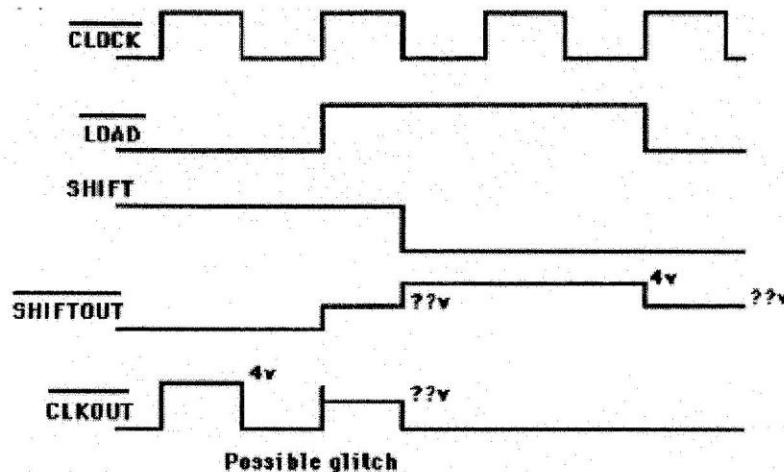
- 4.39 (a) 1. The output of the inverter is internally grounded.
 2. The output of the inverter is externally grounded.
 3. The input being driven by the output of the inverter is internally grounded.
 (b) The output of the inverter is shorted to the output of another logic circuit.

- 4.40** (a) Since Z1-4 is essentially floating, the Logic Probe will show an indeterminate logic level.
 (b) There will be 1.4V-1.8V at the output.



Terminal Z2-9 will be floating (HIGH in TTL) since Z1-4 is opened internally. Thus, the signal at Z2-8 is the opposite of the signal at Z2-10.

(d)



- 4.41** IC Z2-2 will be floating and therefore its voltage will fluctuate as it picks up noise. Thus, Z2-3 level will be unpredictable. IC-Z2 may also become overheated and eventually destroy itself.

- 4.42** 1) First isolate Z1-4 from Z2-1 by using one of the following methods:

- (a) cutting the trace from Z1-4 to Z2-1.
- (b) clipping pin 4 of Z1.
- (c) clipping pin 1 of Z2.

2) Check to see if Z1-4 is pulsing. If it is, then one can be sure that the inverter Z1 is working properly. If it's always LOW (internally shorted to ground) then inverter Z1 must be replaced.

3) If step 2 above proves IC Z1 to be working properly then the problem must be with NAND gate Z2 (internally shorted to ground). By using a logic probe, check the logic level at Z2-1. Chances are that it will have a permanent logic LOW which kept Z1-4 LOW and Z2-3 HIGH. Replace Z2.

- 4.43** 1) Faulty IC bias (Vcc and/or Ground).
2) Z2-2 is internally open (floating).
3) Z2-1 is internally open (floating).
4) Z2-3 is internally open (floating).

Procedure: With a VOM or logic probe, check Vcc and Ground to the IC. If the Vcc and Ground measurements are correct, disconnect Z2-3 from any load it may be driving. If problem persists, replace Z2.

- 4.44** Yes. (c), (e), (f).

- (a) No. This would've kept point X at a logic LOW permanently and the first case (A=1, B=0) wouldn't have worked.
- (b) No. An open at Z2-13 has the same effect as a logic HIGH (only in TTL). Thus, in the second case (A=0,B=1,C=1) Z2-11 would've been LOW and Z2-8 HIGH.
- (d) No. This would've cause IC Z2 to be unbiased and prevent the circuit from working properly for the first case.
- (g) No. This would've caused Z2-10 to be always LOW and Z2-8 HIGH for all cases.

- 4.45** 1) Make A=0 (Z1-1), B=1 (Z1-2) and C=1 (Z2-12). This is the case that causes the circuit to malfunction. Note that the other three possible combinations of A and B do not cause a problem. We know that IC Z1 is working from the results of the first case.

- 2) The logic levels at Z2-13 and Z2-12 should be HIGH.

- (a) Check to see if Z2-11 has a logic LOW.
(b) If Z2-11 is LOW and Z2-9 isn't turn off the power to the circuit.
(c) Use a VOM to make a continuity check between Z2-11 and Z2-9. If there is an open, find it and restore the continuity between these two points.

3) If after performing step two the technician finds that there is a good connection between Z2-11 and Z2-9, then one could conclude that either output Z2-11 or input Z2-9 is externally shorted to Vcc. Since the circuit still has the power turned off from the last check, the technician should make a continuity check to see if the trace between Z2-11 and Z2-9 is externally shorted to Vcc. If there is a short to Vcc, find it and eliminate it. If no external short to Vcc is found then either Z2-11 or Z2-9 or both must be internally short to Vcc or have an internal open. In any case the replacement of IC Z2 should be performed.

- 4.46** This is a tough one. You have noticed that Z2-6 and Z2-11 will be at the same logic level except for the two cases that don't work. For those cases, Z2-6 and Z2-11 are supposed to be different. Since they measure indeterminate for those cases, it is likely that Z2-6 and Z2-11 are shorted together, probably by a solder bridge. The short will have no effect for all those cases where these two outputs are at the same level.

- 4.47** (b) If Z1-2 was internally shorted to ground, whenever the passenger failed to fastened his/her seat-belt the circuit would've not detected this ALARM condition.
(c) Since this is a TTL logic circuit, if there was an open connection between Z2-6 and Z2-10, the circuit would've operated as if a logic HIGH was present at Z2-10. This would've caused the circuit to ALWAYS assume that a passenger was in the seat with the respective seat-belt fastened.

- 4.48** Since the problem only manifests itself when an occupant is present in the car and the ignition is turned on, it can be deduced that IC Z2 is working properly. The problem must be with IC Z1. The following are the possible circuit failures:

- (a) IC Z1 is not properly biased. } Most likely
(b) IC Z1 is plugged in backwards. } problems.

Remote possibilities:

- (c) Z1-4 and Z1-2 are internally shorted to Vcc.
(d) Z1-4 and Z1-2 are internally open.
(e) An open connection from Z1-2 to Z2-5, and from Z1-4 to Z2-2.
(f) Connection from Z1-2 to Z2-5 is externally shorted to Vcc as well as the connection from Z1-4 to Z2-2.
(g) Z1-1 and Z1-3 are internally shorted to Ground.

Procedure:

- 1) Make the necessary voltage measurements to confirm proper IC Z1 bias. Check for proper IC Z1 orientation.
- 2) Check the logic levels at Z1-2 and Z1-4 with a logic probe. If IC Z1 is working properly then a TTL logic LOW should be present at these points.
- 3) If these logic levels are still HIGH, by using an ohmmeter check for any external shorts to Vcc or open PC traces.
- 4) Check the logic levels at Z1-1 and Z1-3 with a logic probe. If IC Z1 is to work properly then a TTL logic HIGH should be present at these points.
- 5) If these logic levels are LOW, use an ohmmeter to check for any external shorts to Ground.
- 6) If the above steps do not reveal a probable cause, Z1 must be internally damaged and it must be replaced.

- 4.49** For some reason Z2-13 is always HIGH. The following are the possible circuit failures:

- (a) Z2-13 is internally shorted to Vcc.
(b) Z2-8 is internally shorted to Vcc.
(c) Connection from Z2-8 to Z2-13 is open or externally shorted to Vcc.
(d) Z2-9 or Z2-10 are internally shorted to Ground.
(e) Z2-3 or Z2-6 are internally shorted to Ground.
(f) Connections from Z2-3 to Z2-9 or from Z2-6 to Z2-10 are externally shorted to Ground.

Procedure:

The first troubleshooting step is to make sure that all of the ICs are properly biased (Vcc and Ground) and oriented.

I) Isolate Z2-13 from Z2-8 by cutting the trace on the PC board or by clipping the proper pin on IC Z2 (either pin 8 or pin 13). Check the voltage level at Z2-13 with a VOM. It should be about 0v since it's floating at this point. If the voltage is \approx Vcc, Z2-13 is either internally or externally shorted to Vcc and it should be replaced.

II) If a fault is not found after performing step I, then check the logic level at Z2-8 with a logic probe. If it's HIGH, check the logic levels at Z2-9 and Z2-10. One of them or both should be LOW. If they are both HIGH, IC Z2-8 is internally or externally shorted to Vcc.

III) **If Z2-9 is LOW**

Check the logic levels at Z2-1 and Z2-2. They should be both LOW. If they are LOW, isolate Z2-3 from Z2-9 by cutting the trace on the PC board or by clipping the appropriate pin (Z2-3 or Z2-9). Check the logic levels at Z2-3 and Z2-9 with a logic probe. If either input is LOW, one must conclude that IC Z2 pin 3 or pin 9 is externally or internally shorted to ground.

IV) If Z2-10 is LOW, the same test procedure should be used for the connection between Z2-10 and Z2-6.

4.50 (a) True; (b) True; (c) False; (d) False; (e) True

4.51 All text between the characters % % serves as comments.

4.52 Comments in a VHDL design file are indicated by --.

4.53 A special socket that allows you to drop the chip in and then clamp the contacts onto the pins.

4.54 1) Boolean equation; 2) Truth table; 3) Schematic diagram

4.55 JEDEC - Joint Electronic Device Engineering Council; HDL - Hardware Description Language

4.56 (a) AHDL: gadgets[7..0] :OUTPUT;
VHDL gadgets :OUT BIT_VECTOR (7 DOWNTO 0);

(b) AHDL buzzer :OUTPUT;
VHDL buzzer :OUT BIT;

(c) AHDL: altitude[15..0] :INPUT;
VHDL altitude :IN INTEGER RANGE 0 TO 65535);

(d) AHDL VARIABLE
wire2 :NODE;
VHDL SIGNAL wire2 :BIT ;

4.57 (a) AHDL H"98" B"10011000" 152
VHDL X"98" B"10011000" 152

(b) AHDL H"254" B"1001010100" 596
VHDL X"254" B"1001010100" 596

(c) AHDL H"3C4" B"1111000100" 964
VHDL X"3C4" B"1111000100" 964

4.58

```
SUBDESIGN hw
(
    inbits[3..0]      :INPUT;
    outbits[3..0]     :OUTPUT;
)
ENTITY hw IS
Port   (
    inbits          :IN BIT_VECTOR (3 downto 0);
    outbits         :OUT BIT_VECTOR (3 downto 0)
);
END hw;

AHDL           outbits[3]      =      inbits[1];
                outbits[2]      =      inbits[3];
                outbits[1]      =      inbits[0];
                outbits[0]      =      inbits[2];

VHDL           outbits(3)     <=     inbits(1);
                outbits(2)     <=     inbits(3);
                outbits(1)     <=     inbits(0);
                outbits(0)     <=     inbits(2);
```

4.59

TABLE

(a,b,c)	=>	y;
(0,0,0)	=>	0;
(0,0,1)	=>	0;
(0,1,0)	=>	1;
(0,1,1)	=>	1;
(1,0,0)	=>	1;
(1,0,1)	=>	0;
(1,1,0)	=>	1;
(1,1,1)	=>	1;

END TABLE;

4.60

```
BEGIN
    IF digital_value[] < 10 THEN
        z = VCC;                      --output a 1
    ELSE z = GND;                     --output a 0
    END IF;
END;
```

4.61

```
WITH in_bits SELECT
    y      <=
        '0' WHEN "000",
        '0' WHEN "001",
        '1' WHEN "010",
        '1' WHEN "011",
        '1' WHEN "100",
        '0' WHEN "101",
        '1' WHEN "110",
        '1' WHEN "111";
```

4.62

```
PROCESS (digital_value)
BEGIN
    IF (digital_value < 10) THEN z <= '1';
    ELSE z <= '0';
    END IF;
END PROCESS;
```

4.63

```
% Problem 4-63 in AHDL
Digital Systems 10th ed
Neal Widmer
%
SUBDESIGN PROB4_63
(
```

```
    digital_value[3..0] :INPUT;
    y                  :OUTPUT;           --define inputs to block
                                         --define block output
)
```

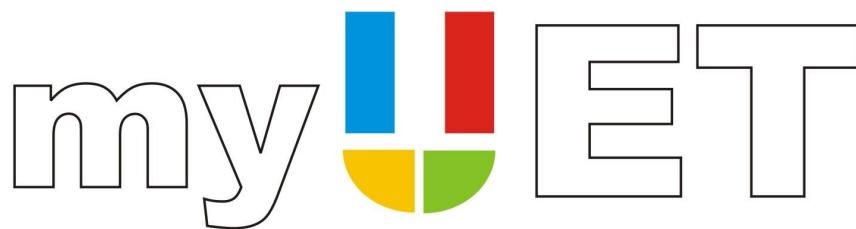
```
BEGIN
    IF digital_value[] > 5 & digital_value[] < 12 THEN
        y = vcc;                         --output a 1
    ELSE y = gnd;                      --output a 0
    END IF;
END;
```

-- NOTE: The digital_value[0] term drops out when this is simplified.
-- The compiler will issue a warning to this effect.

4.63 (in VHDL)

-- USING PROCESS.
-- Digital Systems 10th ed
-- Tocci Widmer Moss

```
ENTITY prob4_63 IS
PORT( digital_value :IN INTEGER RANGE 0 TO 15;          --declare 4-bit input
      z            :OUT BIT);
END fig4_55;
```



```

ARCHITECTURE truth OF fig4_55 IS
BEGIN
    PROCESS (digital_value)
    BEGIN
        IF (digital_value > 5) AND digit_value < 12) THEN
            z <= '1';
        ELSE
            z <= '0';
        END IF;
    END PROCESS ;
END truth;

```

-- NOTE: The digital_value[0] term drops out when this is simplified.
-- The compiler will issue a warning to this effect.

4.64 (a)

```

SUBDESIGN fig4_60
(
    a, b, c      :INPUT;           --define inputs to block
    y             :OUTPUT;          --define outputs
)
VARIABLE
status[2..0]   :NODE;           --holds state of cold, moderate, hot
BEGIN
    status[] = (a, b, c);         --link input bits in order
    CASE status[] IS
        WHEN b"010"      => y = VCC;
        WHEN b"011"      => y = VCC;
        WHEN b"111"      => y = VCC;
        WHEN OTHERS       => y = GND;
    END CASE;
END;

```

4.64 (b)

```

ENTITY fig4_61 IS
port(
    a, b, c      :IN bit;          --declare 3 bits input
    y             :OUT BIT);
END fig4_61;

ARCHITECTURE copy OF fig4_61 IS
SIGNAL status      :BIT_VECTOR (2 downto 0);
BEGIN
    status <= a & b & c;           --link bits in order.
    PROCESS (status)
    BEGIN
        CASE status IS
            WHEN "010" => y <= '1';
            WHEN "011" => y <= '1';
            WHEN "111" => y <= '1';
            WHEN OTHERS => y <= '0';
        END CASE;
    END PROCESS ;
END copy;

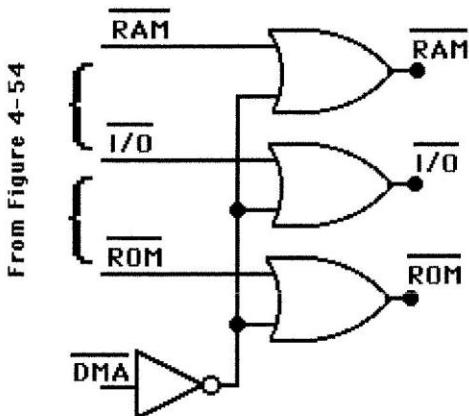
```

4.65 $S = \overline{P} \# (Q \& R)$

4.66 $P = D_3 \$ D_2 \$ D_0 \$ D_1$

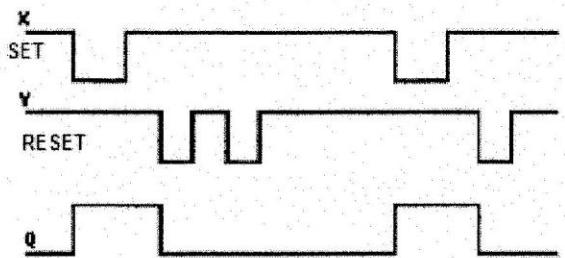
- 4.67**
- (a) Two-dimensional form of a truth table used to simplify a sum-of-products expression.
 - (b) Logic expression consisting of two or more AND terms (products) that are ORed together.
 - (c) Logic circuit that produces an even or odd parity bit for a given set of input data bits.
 - (d) Group of eight 1s that are adjacent to each other within a Karnaugh map.
 - (e) Logic circuit that controls the passage of an input signal through to the output.
 - (f) Situation when a circuit's output level for a given set of input conditions can be assigned as either a 1 or 0.
 - (g) Input signal that is left disconnected in a logic circuit.
 - (h) Whenever a logic voltage level of a particular logic family falls out of the required range of voltages for either a logic 0 or logic 1.
 - (i) Signal contention is when two signals are "fighting" each other.
 - (j) Programmable Logic Device
 - (k) The TTL (Transistor-Transistor-Logic) family is the major family of bipolar digital ICs.
 - (l) The CMOS (Complementary Metal Oxide Semiconductor) family belongs to the class of unipolar digital ICs.
- 4.68** RAM } $00000000_2 - 11101111_2 = 00_{16} - EF_{16}$
 I/O } $11110000_2 = F0_{16}$
 ROM } $11110001_2 - 11111111_2 = F1_{16} - FF_{16}$

4.69



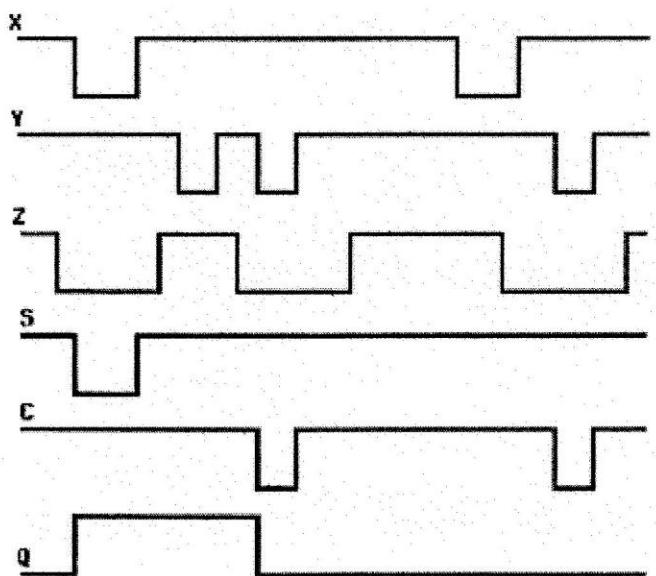
CHAPTER FIVE - Flip-Flops and Related Devices

5.1

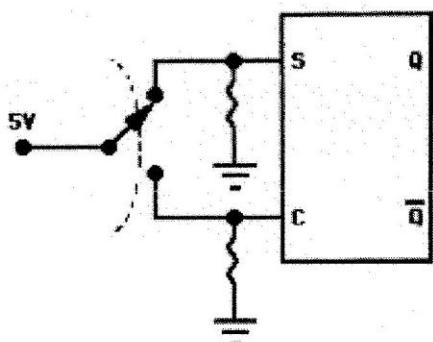


5.2 Same Q output as 5.1.

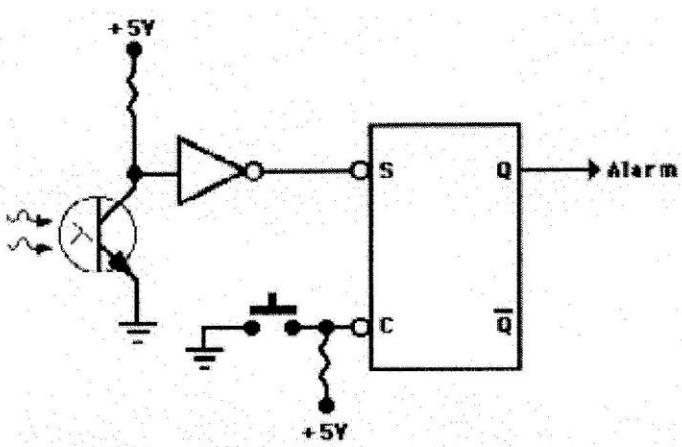
5.3



5.4



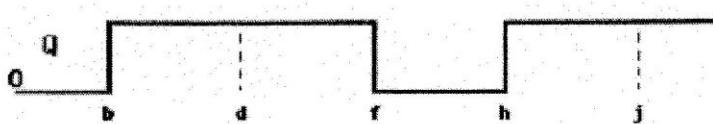
5.5 One possibility:



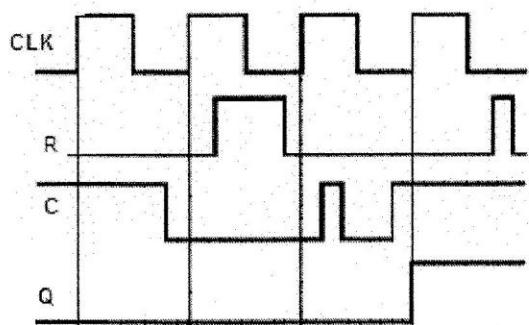
5.6 The response shown would occur if the NAND latch is not working as a Flip-Flop. A permanent logic HIGH at IC Z1-4 will prevent the latch from working properly and therefore the switch bounce will appear at Z1-6. When the 1 KHz squarewave is high, the switch bounce will be present at Z2-6.

5.7 Control inputs have to be stable for $t_S=20\text{ns}$ prior to the clock transition.

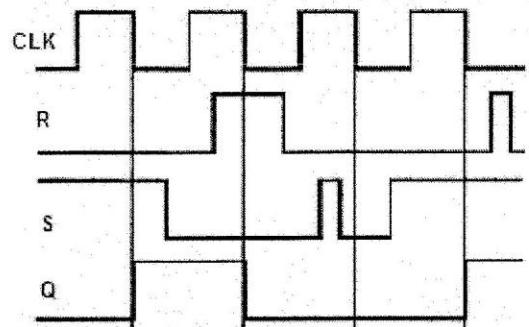
5.8 The FF will respond at times b, d, f, h, j corresponding to negative-going CLK transitions.



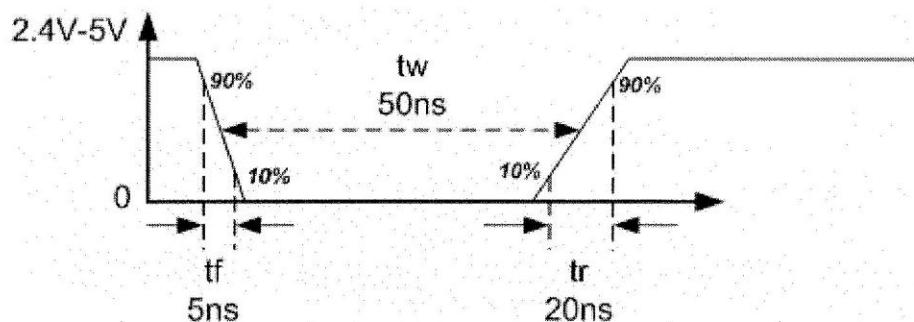
5.9 Assuming that Q=0 initially (for the positive edge triggered S-C FF).



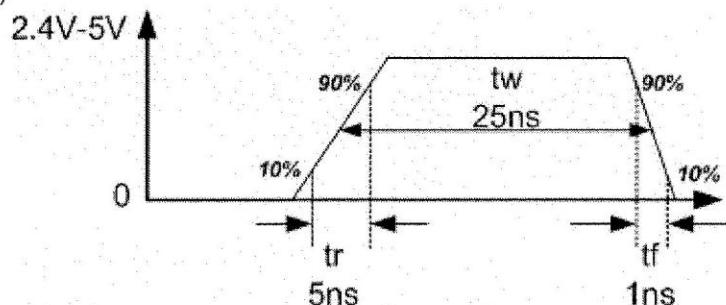
Assuming that Q=0 initially (for the negative edge triggered S-C FF).



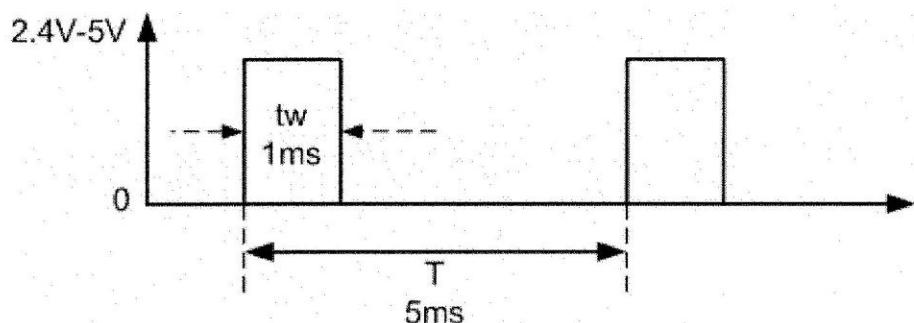
5.10 (a)



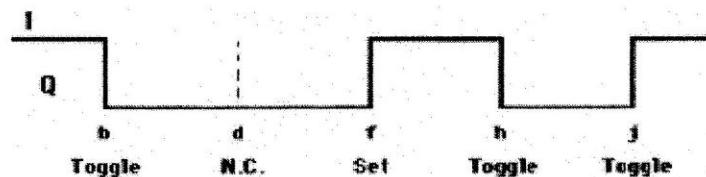
(b)



(c)



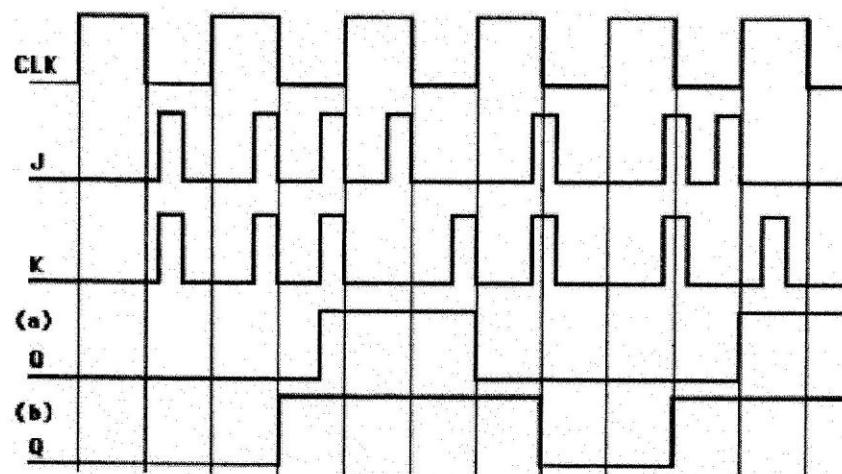
5.11 FF can change state only at points b, d, f, h, j based on values of J and K inputs.



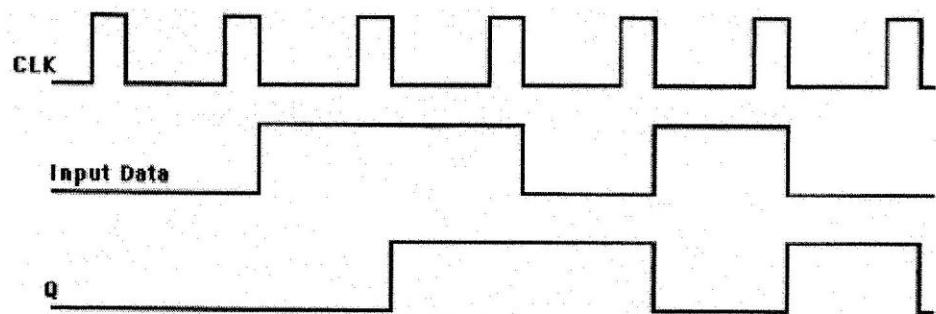
5.12 (a) Connect the J and K inputs permanently HIGH. The Q output will be a squarewave with a frequency of 5 KHz.

(b) The Q output will be a squarewave with a frequency of 2.5 KHz.

5.13

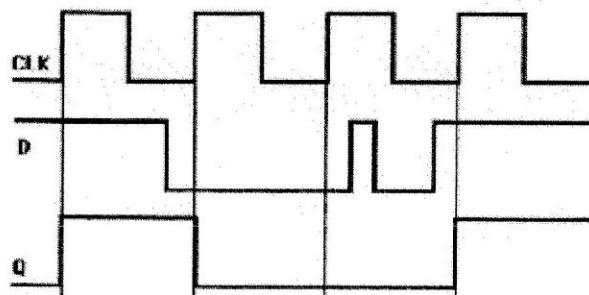


- 5.14 (a) Since the FF has $t_H=0$, the FF will respond to the value present on the D input just prior to the NGT of the clock.

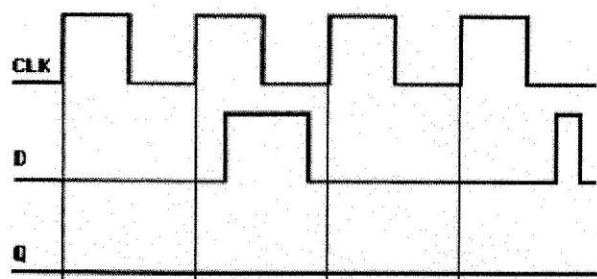


- (b) Connect Q to the D input of a second FF, and connect the clock signal to the second FF. The output of the second FF will be delayed by 2 clock periods from the Input Data.

5.15 (a)

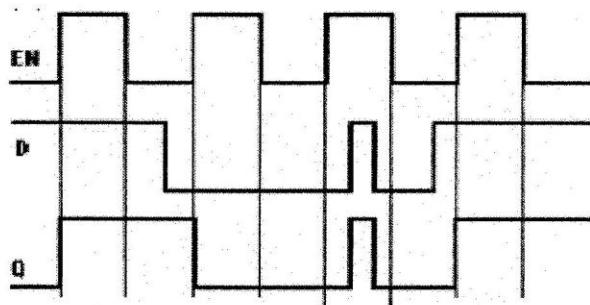


(b)

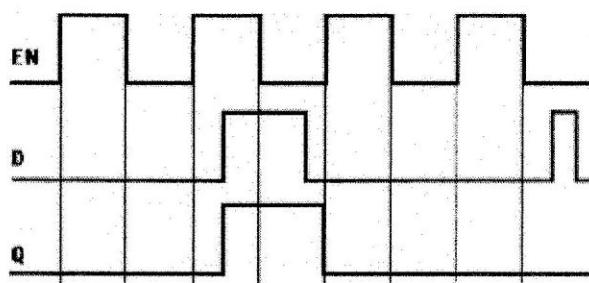


5.16 Q is a 500 Hz square wave.

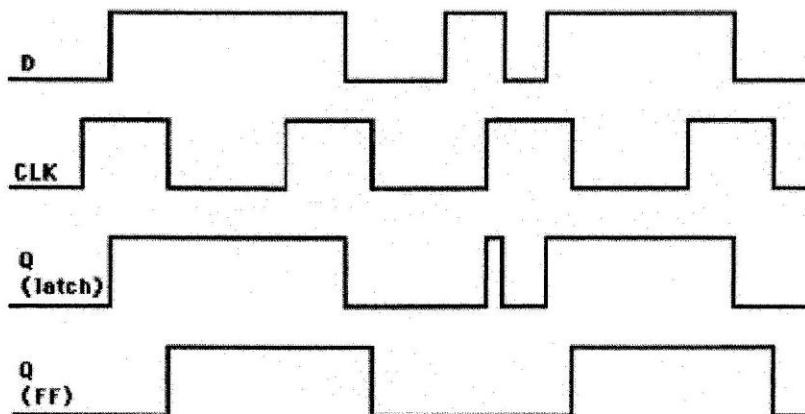
5.17 (a)



(b)

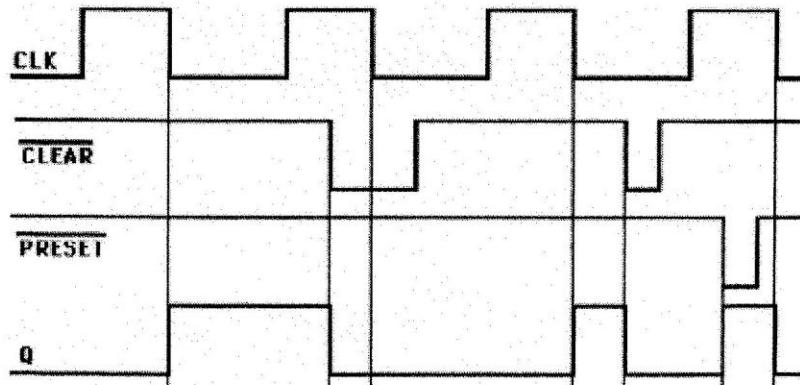


5.18

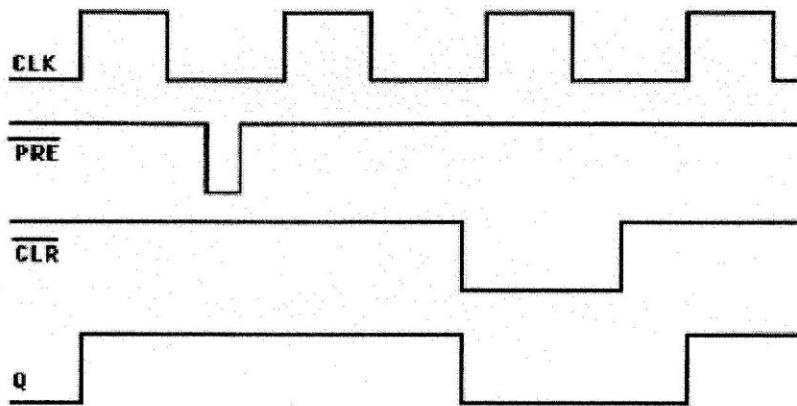


5.19 If \bar{Q} is connected back to D, the Q and \bar{Q} outputs will oscillate while CLK is HIGH. This is because $\bar{Q} = 1$ will produce S=0, C=1 which will make $\bar{Q} = 0$. This $\bar{Q} = 0$ then will make S=1, C=0 which will make $\bar{Q} = 1$.

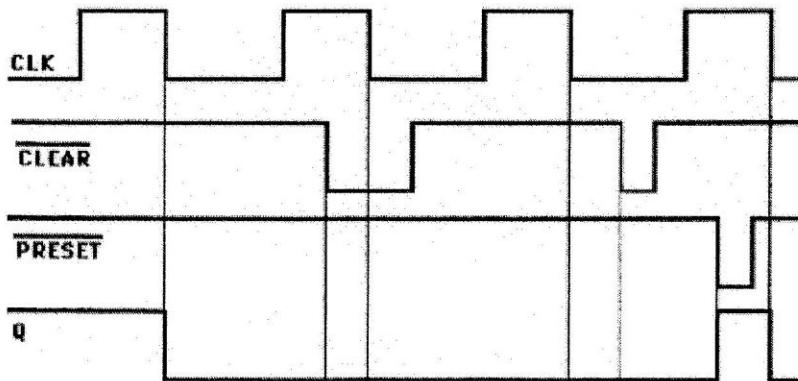
5.20 J=K=1 so FF will toggle on each CLK negative-going edge, unless either PRESET or CLEAR inputs is LOW.



5.21



5.22



- 5.23** (a) t_{pLH} from CLK to Q is 200ns.
 (b) With a t_H = 5ns, the 7474 requires its control inputs to remain stable the longest time after the CLK transition.
 With a t_S = 60ns, the 74C74 requires its control inputs to remain stable the longest time before the CLK transition.
 (c) t_{W(L)} at PRE is 30ns.

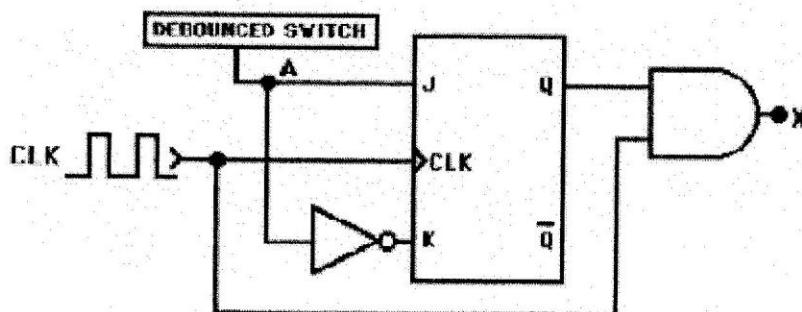
- 5.24** (a) t_{pHL}, CLR-Q = 24ns
 (b) t_{pLH}, PRE-CLR-Q = 41ns

$$(c) T_{min} = \frac{1}{F_{max}} = \frac{1}{15MHz} = 66.7ns$$

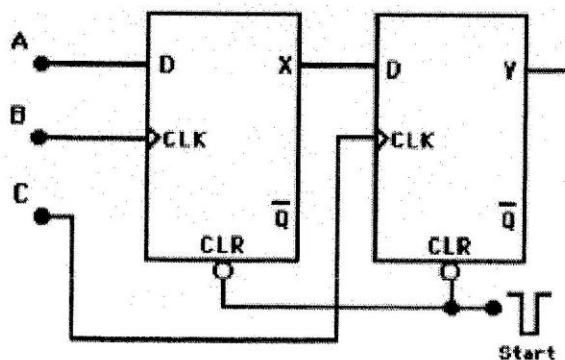
(d) t_{su(min)} = 25ns. No. There is insufficient time.

(e) t_{pLH} = 25ns CLR to Q

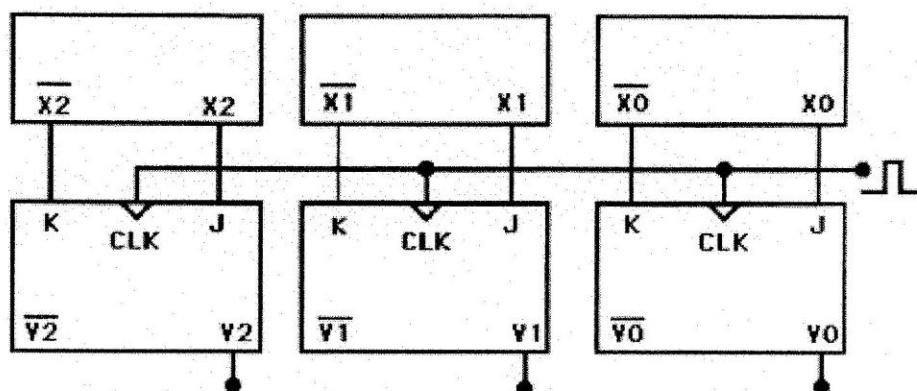
5.25



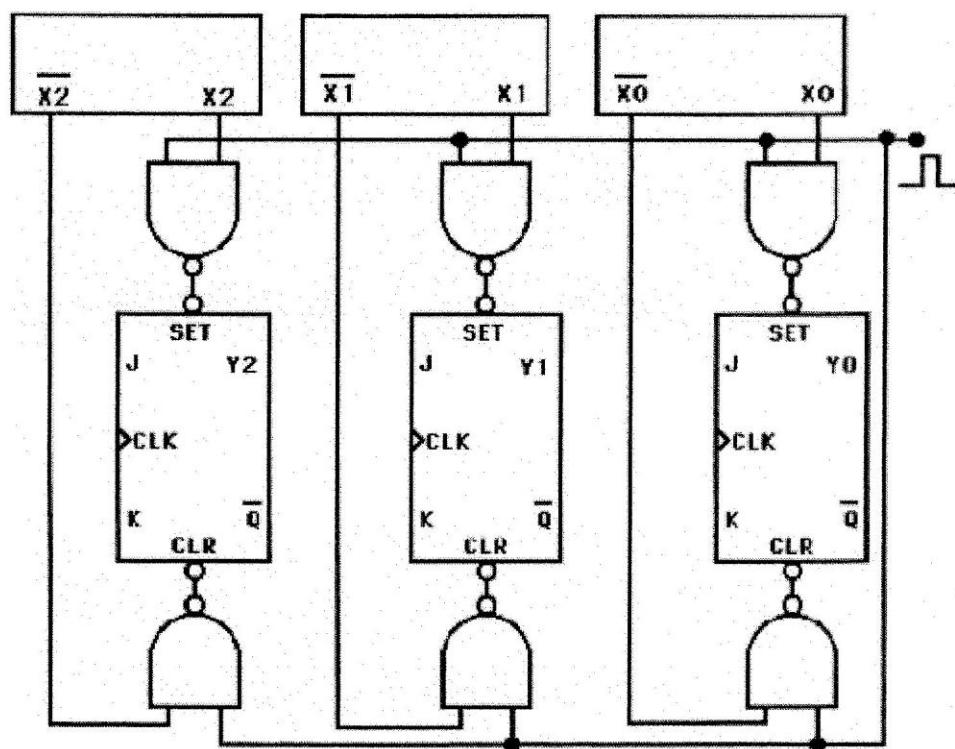
- 5.26** (a) Y can go HIGH only when C goes HIGH while X is already HIGH. X can go HIGH only if B goes HIGH while A is HIGH. Thus, the correct sequence is A,B,C.
 (b) The START pulse initially clears X and Y to 0 before applying the A,B,C signals.
 (c)



5.27 (a)



(b)



- 5.28 In this arrangement, the data shifts accordingly:



X3 X2 X1 X0

I	0	1	1	Initial State
1	1	0	1	Clock Pulse 1
1	1	1	0	Clock Pulse 2
0	1	1	1	Clock Pulse 3
1	0	1	1	Clock Pulse 4
1	1	0	1	Clock Pulse 5
1	1	1	0	Clock Pulse 6
0	1	1	1	Clock Pulse 7
1	0	1	1	Clock Pulse 8

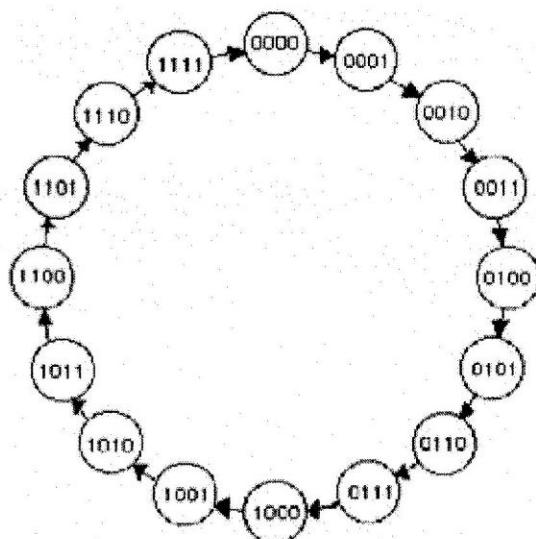
- 5.29 Connect outputs X0 to D input of FF X2 so that the contents of the X register will be recirculated.

- 5.30 This is a counter that will recycle every 8 pulses (MOD 8 counter).

(a) Count after 13 clock pulses is 5 (101); Count after 99 clock pulses is 3 (011); Count after 256 clock pulses is 0 (000).

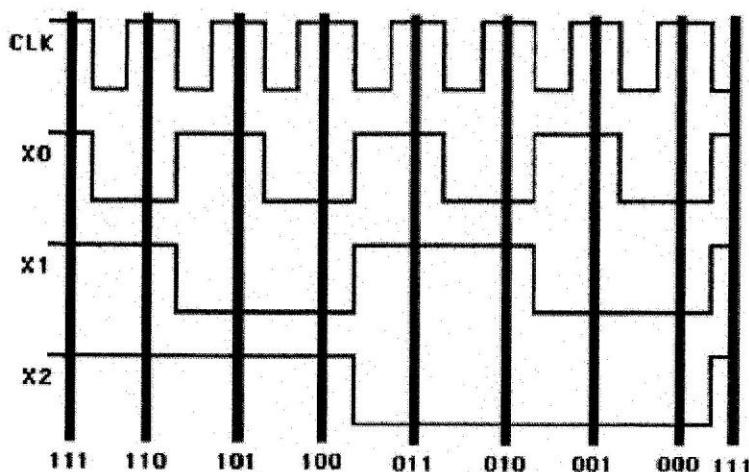
(b) Count after 13 clock pulses is 1 (001); Count after 99 clock pulses is 7 (111); Count after 256 clock pulses is 4 (100).

(c) State diagram for a MOD-16 counter

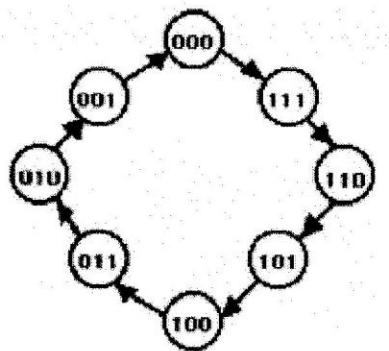


If the input frequency is 80 MHz the output waveform at X3 will be a squarewave with a frequency of 500 KHz (80 MHz/16).

5.31



5.32



5.33 (a) $2^N-1=1023$, so that $2^N=1024$. Thus, $N=10$ flip-flops.

(b) With N FFs, the MOD-number is $2^N=1024$ so that the frequency division at the last FF will be $1/1024$ relative to the input clock. Thus, output frequency = $2\text{MHz}/1024 = 1953 \text{ Hz}$.

(c) MOD-number= $2^N=1024$.

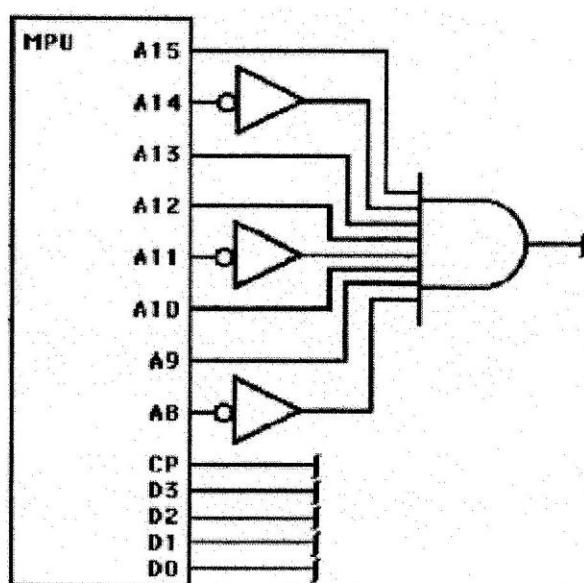
(d) Every 1024 pulses the counter recycles through zero. Thus, after 2048 pulses the counter is back at count zero. Therefore, after 2060 pulses the counter will be at count 12 (i.e. $1024 + 1024 + 12 = 2060$).

5.34 (a) MOD-number = $256 \text{ KHz}/2\text{KHz} = 128$.

(b) $128=2^N$. The maximum count is $2^N-1=127$. Thus, the range is 0 to 127.

5.35 The counter recycled back to 00000000 after $2^8=256$ customers.

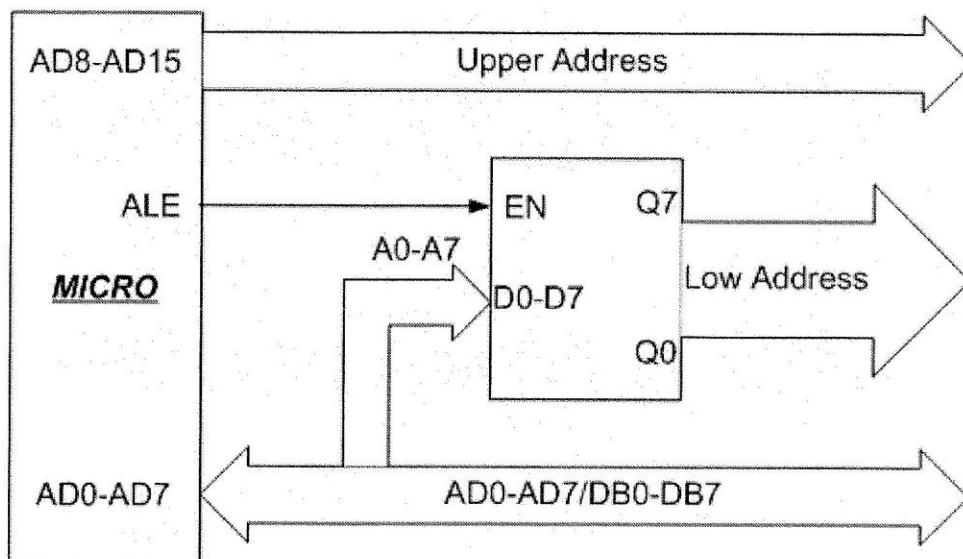
5.36



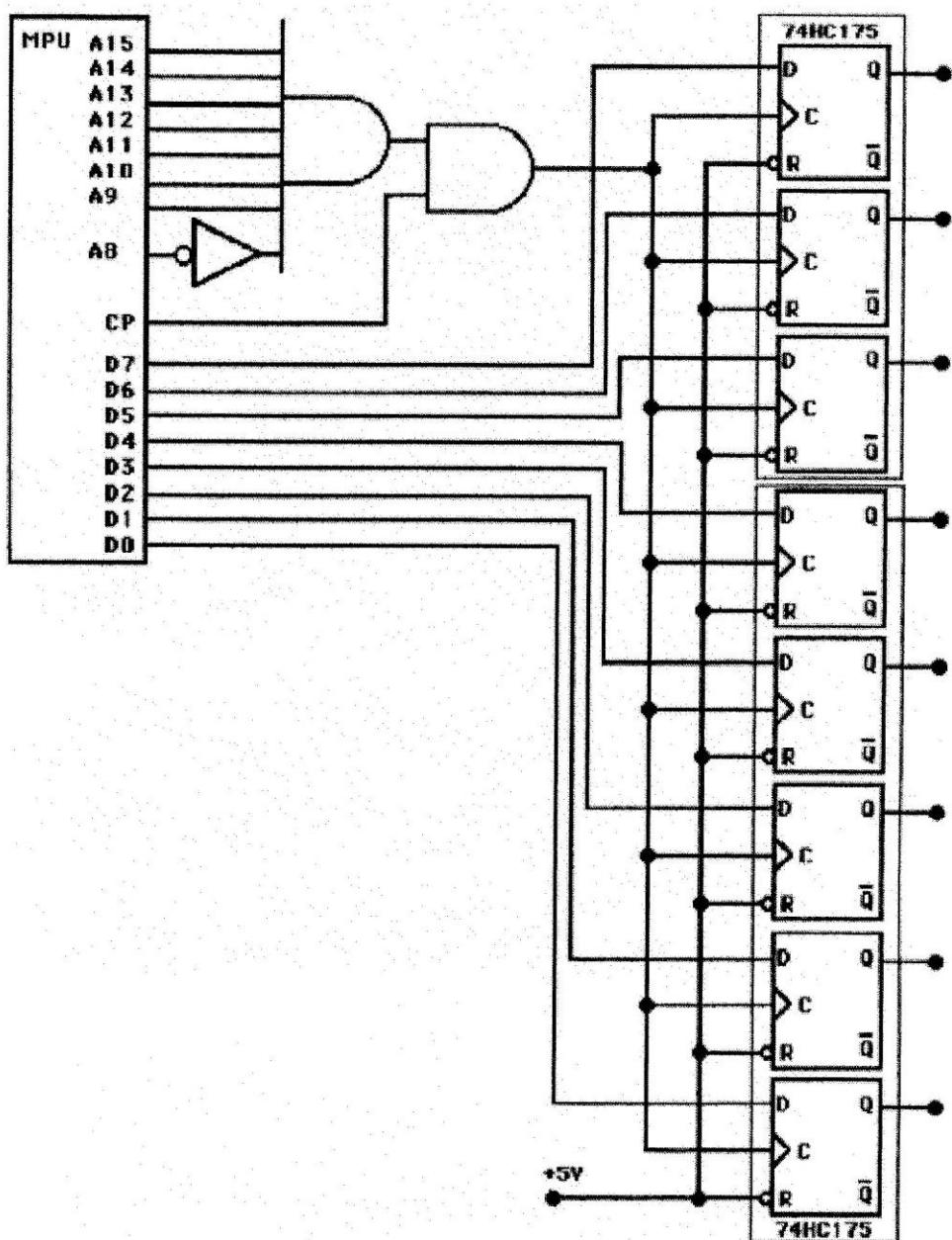
- 5.37** Regardless of the logic state of the address line A8, data gets transferred from the MPU to the X register. Thus, the problem is in the connection between the address line A8 from the MPU and the 8-input AND gate. The following are some of the circuit faults that could cause this malfunction:

- (a) External open on address line A8 between the MPU and the input of the Inverter.
- (b) External short to Vcc on address line A8 between the MPU and the input of the Inverter.
- (c) External open on the line connecting the output of the Inverter and the input of the AND gate.
- (d) External short to Vcc on the line connecting the output of the Inverter and the input of the AND gate.
- (e) Internal open or short to Vcc on the input of the Inverter.
- (f) Internal open or short to Vcc on the output of the Inverter.
- (g) Internal open or short to Vcc on the input of the AND gate.

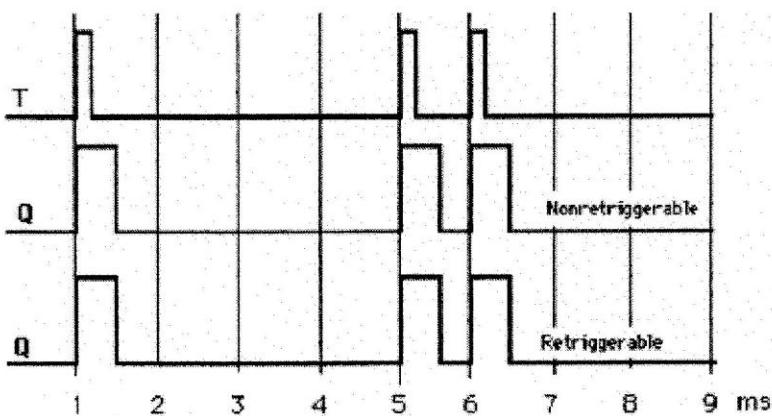
5.38



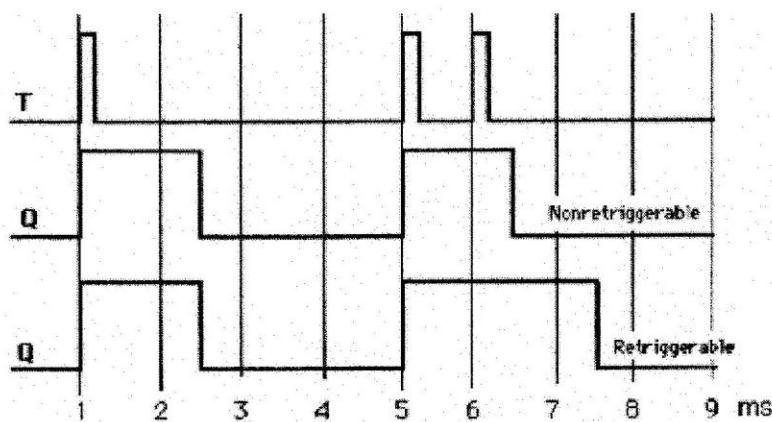
5.39



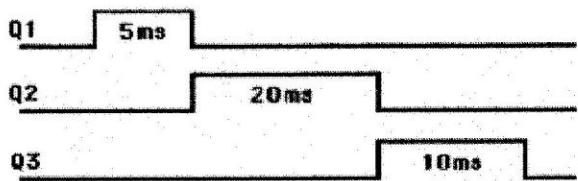
5.40 With $t_p = 0.5\text{ms}$



With $t_p = 1.5\text{ms}$



5.41



5.42 (a) Closing S_1 clears X to 0. Since the OS has $t_p=1\text{ms}$, the OS will be triggered before the end of the t_p interval for frequencies greater than 1 KHz. Thus, \bar{Q} will stay LOW.

(b) If the input frequency drops below 1 KHz, the \bar{Q} will return HIGH before the OS is triggered again. This PGT at \bar{Q} will clock X to the 1 state.

(c) Change t_p to $1/50\text{ KHz} = 20\mu\text{s}$.

5.43 (a) A_1 or A_2 has to be LOW, and a PGT must occur at B .

(b) B and A_2 have to be HIGH, and a NGT must occur at A_1 .

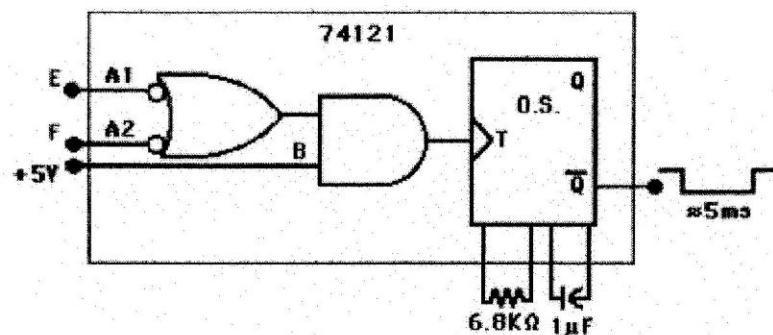
5.44 (a) One possibility:

$$0.7 R_T C_T = 5\text{ms}$$

$$\text{Let } C_T = 1\mu\text{F}; 0.7 R_T = 5\text{ms}/1\mu\text{F} = 5000$$

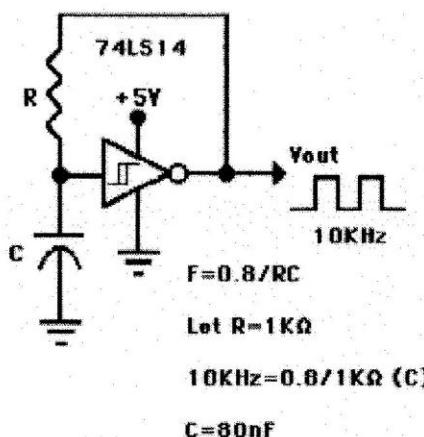
$$R_T = 7143\Omega \approx 6.8\text{K}\Omega \text{ (std. value).}$$

If an accurate 5ms is required, an adjustable R_T should be used.



(b) Connect G to input B of 74121.

5.45



5.46 One possibility:

$$F=40\text{ KHz}; T=25\mu\text{s}; t_1=t_2=12.5\mu\text{s}$$

For a squarewave $\mathbf{RA} \ll \mathbf{RB}$; Let $RA=1\text{K}\Omega$ and $RB=10\text{K}\Omega$

$$t_1=0.693(RB)(C); 12.5\mu\text{s}=0.693(10\text{K}\Omega)(C); C=1800\text{pF}$$

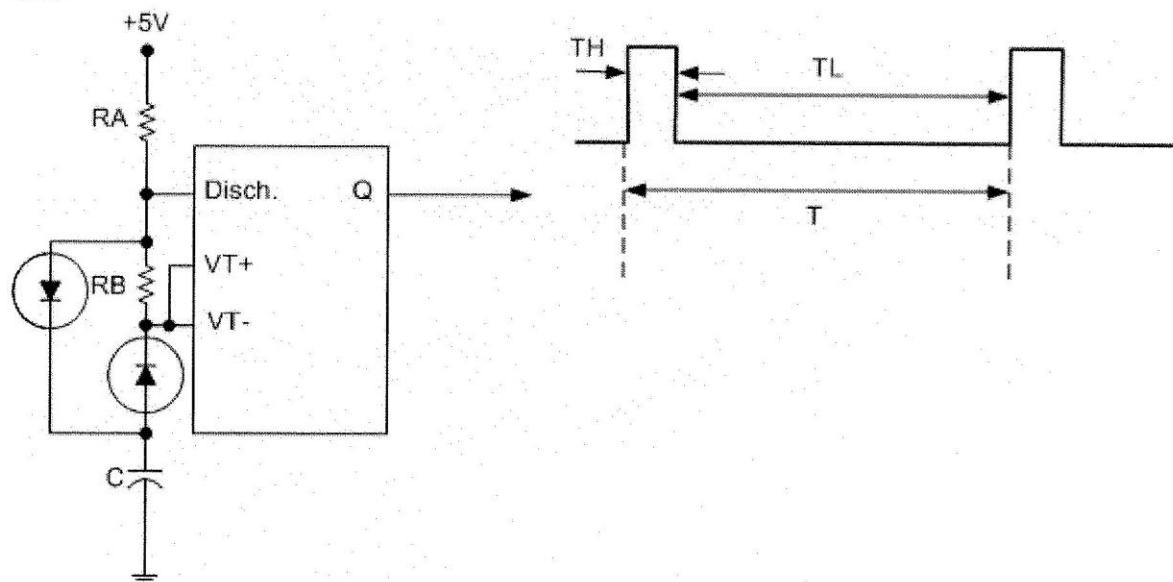
$$T=0.693(RA+2RB)C; T=0.693(1\text{K}\Omega+20\text{K}\Omega)1800\text{pF}$$

$$T=26.2\mu\text{s}; F=1/T; F=38\text{ KHz} \text{ (almost squarewave).}$$

5.47 One possibility:

Reduce by half the 1800pF. This will create a $T=13.1\mu\text{s}$ or $F=76.35\text{ KHz}$ (almost square wave). Now, take the output of the 555 Timer and connect it to the CLK input of a J-K FF wired in the toggle mode (J and K inputs connected to +5V). The result at the Q output of the J-K FF is a perfect 38.17 KHz square wave.

5.48



$$T = \frac{1}{F} = \frac{1}{5\text{KHz}} = 200\mu\text{s}$$

$$TH = (10\%)(200\mu\text{s}) = 20\mu\text{s}$$

$$TL = T - TH = 200\mu\text{s} - 20\mu\text{s} = 180\mu\text{s}$$

Choose $C = 0.01\mu\text{F}$

$$TL = 0.75(RB)(C)$$

$$RB = \frac{TL}{0.75C} = \frac{180\mu\text{s}}{0.75(0.01\mu\text{F})} = 13.5\text{K}\Omega$$

$$RA = \frac{TH}{0.75C} = \frac{20\mu\text{s}}{0.75(0.01\mu\text{F})} = 1.5\text{K}\Omega$$

5.49 (a)

