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

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

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***

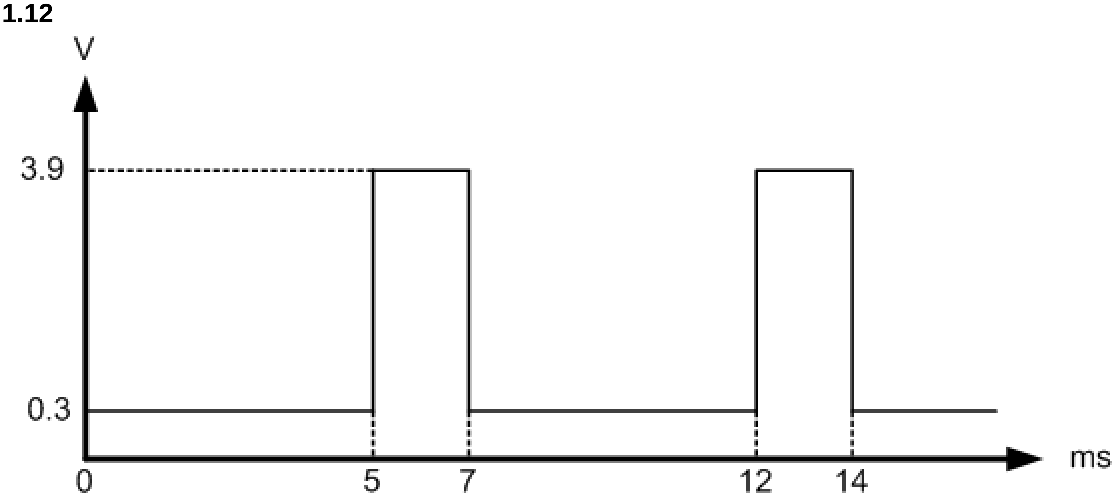
1.8 2^N-1 = 2^{14}-1 = ***16,383***

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) $10110_2 = 16+4+2 = \boldsymbol{22_{10}}$
(b) $10010101_2 = 128+16+4+1 = \boldsymbol{149_{10}}$
(c) $100100001001_2 = 2048+256+8+1 = \boldsymbol{2313_{10}}$
(d) $01101011_2 = 64+32+8+2+1 = \boldsymbol{107_{10}}$
(e) $11111111_2 = 128+64+32+16+8+4+2+1 = \boldsymbol{255_{10}}$
(f) $01101111_2 = 64+32+8+4+2+1 = \boldsymbol{111_{10}}$
(g) $1111010111_2 = 512+256+128+64+16+4+2+1 = \boldsymbol{983_{10}}$
(h) $11011111_2 = 128+64+16+8+4+2+1 = \boldsymbol{223_{10}}$
(i) $100110_2 = 32+4+2 = \boldsymbol{38_{10}}$
(j) $1101_2 = 8+4+1 = \boldsymbol{13_{10}}$
(k) $111011_2 = 32+16+8+2+1 = \boldsymbol{59_{10}}$
(l) $1010101_2 = 64+16+4+1 = \boldsymbol{85_{10}}$

2.2
(a) $37_{10} = 32+4+1 = \boldsymbol{100101_2}$
(b) $13_{10} = 8+4+1 = \boldsymbol{1101_2}$
(c) $189_{10} = 128+32+16+8+4+1 = \boldsymbol{10111101_2}$
(d) $1000_{10} = 512+256+128+64+32+8 = \boldsymbol{1111101000_2}$
(e) $77_{10} = 64+8+4+1 = \boldsymbol{1001101_2}$
(f) $390_{10} = 256+128+4+2 = \boldsymbol{110000110_2}$
(g) $205_{10} = 128+64+8+4+1 = \boldsymbol{11001101_2}$
(h) $2133_{10} = 2048+64+16+4+1 = \boldsymbol{100001010101_2}$
(i) $511_{10} = 256+128+64+32+16+8+4+2+1 = \boldsymbol{111111111_2}$
(j) $25_{10} = 16+8+1 = \boldsymbol{11001_2}$
(k) $52_{10} = 32+16+4 = \boldsymbol{110100_2}$
(l) $47_{10} = 32+8+4+2+1 = \boldsymbol{101111_2}$

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

2.4
(a) $743_{16} = 7\text{x}16^2+4\text{x}16^1+3\text{x}16^0 = \boldsymbol{1859_{10}}$
(b) $36_{16} = 3\text{x}16^1+6\text{x}16^0 = \boldsymbol{54_{10}}$
(c) $37FD_{16} = 3\text{x}16^3+7\text{x}16^2+15\text{x}16^1+13\text{x}16^0 = \boldsymbol{14333_{10}}$
(d) $2000_{16} = 2\text{x}16^3 = \boldsymbol{8192_{10}}$
(e) $165_{16} = 1\text{x}16^2+6\text{x}16^1+5\text{x}16^0 = \boldsymbol{357_{10}}$
(f) $ABCD_{16} = 10\text{x}16^3+11\text{x}16^2+12\text{x}16^1+13\text{x}16^0 = \boldsymbol{43981_{10}}$
(g) $7FF_{16} = 7\text{x}16^2+15\text{x}16^1+15\text{x}16^0 = \boldsymbol{2047_{10}}$
(h) $1204_{16} = 1\text{x}16^3+2\text{x}16^2+4\text{x}16^0 = \boldsymbol{4612_{10}}$
(i) $E71_{16} = 14\text{x}16^3+7\text{x}16^2+1\text{x}16^0 = \boldsymbol{3697_{10}}$
(j) $89_{16} = 8\text{x}16^2+9\text{x}16^0 = \boldsymbol{137_{10}}$
(k) $58_{16} = 5\text{x}16^2+8\text{x}16^0 = \boldsymbol{88_{10}}$
(l) $72_{16} = 7\text{x}16^2+2\text{x}16^0 = \boldsymbol{114_{10}}$

2.5	(a)	59/16 = 3	Remainder of 11 (B)}	
		3/16 = 0	Remainder of 3 }	59_{10} = ***$3B_{16}$***
	(b)	372/16 = 23	Remainder of 4 }	
		23/16 = 1	Remainder of 7 }	
		1/16 = 0	Remainder of 1 }	372_{10} = ***174_{16}***
	(c)	919/16 = 57	Remainder of 7 }	
		57/16 = 3	Remainder of 9 }	
		3/16 = 0	Remainder of 3 }	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 (F)}	
		15/16 = 0	Remainder of 15 (F)}	255_{10} = ***FF_{16}***
	(i)	29/16 = 1	Remainder of 13 (D)}	
		1/16 = 0	Remainder of 1 }	29_{10} = ***$1D_{16}$***
	(j)	33/16 = 2	Remainder of 1 }	
		2/16 = 0	Remainder of 2 }	33_{10} = ***21_{16}***
	(k)	100/16 = 6	Remainder of 4 }	
		6/16 = 0	Remainder of 6 }	100_{10} = ***64_{16}***
	(l)	200/16 = 12	Remainder of 8 }	
		12/16 = 0	Remainder of 12 (C)}	200_{10} = ***$C8_{16}$***

2.6 (a) 743_{16} = ***11101000011_2***
(b) 36_{16} = ***110110_2***
(c) $37FD_{16}$ = ***11011111111101_2***
(d) 2000_{16} = ***10000000000000_2***
(e) 165_{16} = ***101100101_2***
(f) $ABCD_{16}$ = ***1010101111001101_2***
(g) $7FF_{16}$ = ***011111111111_2***
(h) 1204_{16} = ***1001000000100_2***
(i) $E71_{16}$ = ***111001110001_2***
(j) 89_{16} = ***10001001_2***
(k) 58_{16} = ***01011000_2***
(l) 72_{16} = ***01110010_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}***
(i) 100110_2=***26_{16}***
(j) 1101_2=***D_{16}***
(k) 111011_2=***$3B_{16}$***
(l) 1010101_2=***55_{16}***

2.8 195, 196, 197, 198, 199, 19A, 19B, 19C, 19D, 19E, 19F, 1A0….1AF, 1B0....1BF, 1C0....1CF, 1D0....1DF, 1E0....1EF, 1F0....1FF, 200....208.

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; $16^4 \geq 65{,}536$, this is greater than 40,000, so N=4.

2.11 (a) $92_{16} = 9\text{x}16^1+2\text{x}16^0$ = ***146_{10}***
(b) $1A6_{16} = 1\text{x}16^2+10\text{x}16^1+6\text{x}16^0$ = ***422_{10}***
(c) $37FD_{16} = 3\text{x}16^3+7\text{x}16^2+15\text{x}16^1+13\text{x}16^0$ = ***14333_{10}***
(d) $ABCD_{16} = 10\text{x}16^3+11\text{x}16^2+12\text{x}16^1+13\text{x}16^0$ = ***43981_{10}***
(e) $000F_{16} = 0\text{x}16^3+0\text{x}16^2+0\text{x}16^1+15\text{x}16^0$ = ***15_{10}***
(f) $55_{16} = 5\text{x}16^1+5\text{x}16^0$ = ***85_{10}***
(g) $2C0_{16} = 2\text{x}16^2+12\text{x}16^1+0$ = ***704_{10}***
(h) $7FF_{16} = 7\text{x}16^2+15\text{x}16^1+15\text{x}16^0$ = ***2047_{10}***
(i) $19_{16} = 1\text{x}16^1+9\text{x}16^0$ = ***25_{10}***
(j) $42_{16} = 4\text{x}16^1+2\text{x}16^0$ = ***66_{10}***

(k) $CA_{16} = 12 \times 16^1 + 10 \times 16^0 = \boldsymbol{202_{10}}$

(l) $F1_{16} = 15 \times 16^1 + 1 \times 16^0 = \boldsymbol{241_{10}}$

2.12

(a)	75/16	= 4	Remainder of 11 [B]	}
	4/16	= 0	Remainder of 4	} 75_{10}=$\boldsymbol{4B_{16}}$
(b)	314/16	= 19	Remainder of 10 [A]	}
	19/16	= 1	Remainder of 3	}
	1/16	= 0	Remainder of 1	} 314_{10}=$\boldsymbol{13A_{16}}$
(c)	2048/16	= 128	Remainder of 0	}
	128/16	= 8	Remainder of 0	}
	8/16	= 0	Remainder of 8	} 2048_{10}=$\boldsymbol{800_{16}}$
(d)	24/16	= 1	Remainder of 8	
	1/16	= 0	Remainder of 1	} 24_{10}=$\boldsymbol{18_{16}}$
(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}=$\boldsymbol{1C4D_{16}}$
(f)	498/16	= 31	Remainder of 2	}
	31/16	= 1	Remainder of 15 [F]	}
	1/16	= 0	Remainder of 1	} 498_{10}=$\boldsymbol{1F2_{16}}$
(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}=$\boldsymbol{6413_{16}}$
(h)	4095/16	= 255	Remainder of 15 [F]	}
	255/16	= 15	Remainder of 15 [F]	}
	15/16	= 0	Remainder of 15 [F]	} 4095_{10}=$\boldsymbol{FFF_{16}}$
(i)	95/16	= 5	Remainder of 15 [F]	}
	5/16	= 0	Remainder of 5	} 95_{10}=$\boldsymbol{5F_{16}}$
(j)	89/16	= 5	Remainder of 9	}
	5/16	= 0	Remainder of 5	} 89_{10}=$\boldsymbol{59_{16}}$
(k)	128/16	= 8	Remainder of 0	}
	8/16	= 0	Remainder of 8	} 128_{10}=$\boldsymbol{80_{16}}$
(l)	256/16	= 16	Remainder of 0	}
	16/16	= 1	Remainder of 0	}
	1/16	= 0	Remainder of 1	} 256_{10}=$\boldsymbol{100_{16}}$

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 FFF_{16}= $\boldsymbol{4096_{10}}$

2.16 (a) 92_{16}=$\boldsymbol{10010010_2}$
(b) $1A6_{16}$=$\boldsymbol{000110100110_2}$
(c) $37FD_{16}$=$\boldsymbol{0011011111111101_2}$
(d) $ABCD_{16}$=$\boldsymbol{1010101111001101_2}$
(e) $000F_{16}$=$\boldsymbol{1111_2}$
(f) 55_{16}=$\boldsymbol{01010101_2}$
(g) $2C0_{16}$=$\boldsymbol{001011000000_2}$
(h) $7FF_{16}$=$\boldsymbol{011111111111_2}$
(i) 19_{16}=$\boldsymbol{11001_2}$
(j) 42_{16}=$\boldsymbol{1000010_2}$
(k) CA_{16}=$\boldsymbol{11001010_2}$
(l) $F1_{16}$=$\boldsymbol{11110001_2}$

2.17 280_{16}, 281_{16}, 282_{16},..... 288_{16}, 289_{16}, $28A_{16}$, $28B_{16}$, $28C_{16}$, $28D_{16}$, $28E_{16}$, $28F_{16}$, 290_{16}, 291_{16},..... 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: $FFFFF_{16} = (16^5-1) = 1{,}048{,}575_{10}$
Therefore, we need ***five*** hex digits to represent decimal numbers up to 1 million.
With *six* hex digits we can represent a decimal number up to: $FFFFFF_{16} = (16^6-1) = 16{,}777{,}215_{10}$
Therefore, we need ***six*** hex digits to represent decimal numbers up to 4 million.

2.19 (a) $47_{10}=\boldsymbol{01000111_{BCD}}$
(b) $962_{10}=\boldsymbol{100101100010_{BCD}}$
(c) $187_{10}=\boldsymbol{000110000111_{BCD}}$
(d) $6727_{10}=\boldsymbol{0110011100100111_{BCD}}$
(e) $13_{10}=\boldsymbol{00010011_{BCD}}$
(f) $529_{10}=\boldsymbol{010100101001_{BCD}}$
(g) $89{,}627_{10}=\boldsymbol{10001001011000100111_{BCD}}$
(h) $1024_{10}=\boldsymbol{0001000000100100_{BCD}}$
(i) $72_{10}=\boldsymbol{01110010_{BCD}}$
(j) $38_{10}=\boldsymbol{00111000_{BCD}}$
(k) $61_{10}=\boldsymbol{01100001_{BCD}}$
(l) $90_{10}=\boldsymbol{10010000_{BCD}}$

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) $1001\ 0111\ 0101\ 0010_{BCD} = \boldsymbol{9752_{10}}$
(b) $0001\ 1000\ 0100_{BCD} = \boldsymbol{184_{10}}$
(c) $0110\ 1001\ 0101_{BCD} = \boldsymbol{695_{10}}$
(d) $0111\ 0111\ 0111\ 0101_{BCD} = \boldsymbol{7775_{10}}$
(e) $0100\ 1001\ 0010_{BCD} = \boldsymbol{492_{10}}$
(f) $0101\ 0101\ 0101_{BCD} = \boldsymbol{555_{10}}$
(g) $0001\ 0111_{BCD} = \boldsymbol{17_{10}}$
(h) $0001\ 0110_{BCD} = \boldsymbol{16_{10}}$
(i) $0111\ 0101_{BCD} = \boldsymbol{75_{10}}$

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 $\boldsymbol{FFFFFFFF_{16}}$.
(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

Char	Hex	Bin	With odd-parity
x ---------->	78	= 111 1000	**1**111 1000 = F8
space ------->	20	= 010 0000	**0**010 0000 = 20
= ---------->	3D	= 011 1101	**0**011 1101 = 3D
space ----->	20	= 010 0000	**0**010 0000 = 20
3 ---------->	3	= 011 0011	**1**011 0011 = B3
* ---------->	2A	= 010 1010	**0**010 1010 = 2A
y ---------->	79	= 111 1001	**0**111 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

Char	Hex	Bin	With even-parity
x ---------->	78	= 111 1000	**0**111 1000 = 78
space ------->	20	= 010 0000	**1**010 0000 = A0
= ---------->	3D	= 011 1101	**1**011 1101 = BD
space ----->	20	= 010 0000	**1**010 0000 = A0
3 ---------->	3	= 011 0011	**0**011 0011 = 33
* ---------->	2A	= 010 1010	**1**010 1010 = AA
y ---------->	79	= 111 1001	**1**111 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***; 65=***e***; 6E=***n***.
Thus, the name of the person is ***Joe Green***.

2.27 (a) $74_{10} = 01110100_{BCD}$ } **1**01110100
(b) $38_{10} = 00111000_{BCD}$ } **0**00111000
(c) $8884_{10} = 1000100010000100_{BCD}$ } **1**1000100010000100
(d) $275_{10} = 001001110101_{BCD}$ } **0**001001110101
(e) $165_{10} = 000101100101_{BCD}$ } **0**000101100101
(f) $9201_{10} = 1001001000000001_{BCD}$ } **1**1001001000000001
(g) $11_{10} = 00010001_{BCD}$ } **1**00010001
(h) $51_{10} = 01010001_{BCD}$ } **0**01010001

2.28 (a) 1001 0101 1000 **0** { parity bit
9 5 8
Since the number of 1s is 5, there is ***no single-bit*** error.

(b) 0100 0111 0110 **0**
4 7 6
Since there are six 1s there is a ***single*** error.

c) 0111 1100 0001 **1**
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) 1000 0110 0010 **1**
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
(d) 10000_2

2.34 (a) 1011_2
(b) 100111_2
(c) 1101_2
(d) 10001111_2

2.35 (a) $777A_{16}$
(b) $999A_{16}$
(c) 1000_{16}
(d) 2001_{16}
(e) $A00_{16}$
(f) $100B_{16}$
(g) 10_{16}
(h) FF_{16}

2.36 (a) 7778_{16}
(b) 9998_{16}
(c) $0FFE_{16}$
(d) $1FFF_{16}$
(e) $9FE_{16}$
(f) 1009_{16}
(g) E_{16}
(h) FD_{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) $000FF_{16}$
(c) 0000_{16} – $07FF_{16}$

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 ($5x10^9$/440,000).

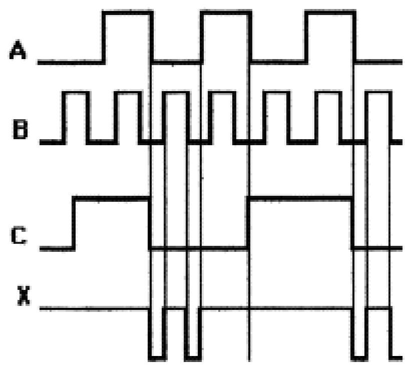
2.39 $254=2^{x}$. Therefore x=7.98˜8-bits

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

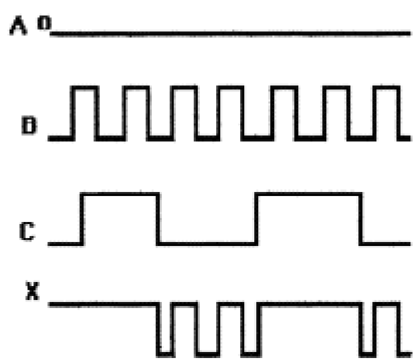
2.41 See Table 2-3 in text.

CHAPTER THREE - Describing Logic Circuits

3.1

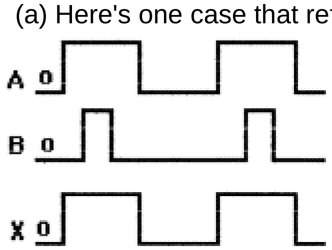


3.2

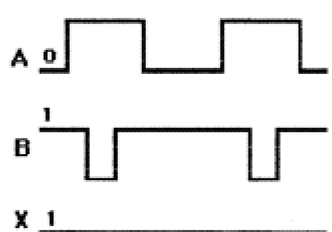


3.3 With A=1, X will always be 1 since the OR gate output is 1 whenever <u>any</u> input is a 1.

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

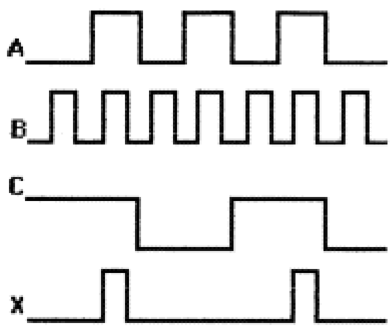


(b) Here's one case that refutes this statement.



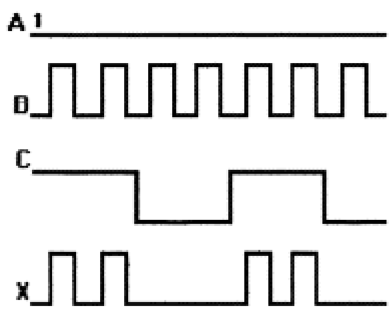
3.5 There are 2^5=32 different input conditions. Only one of these (the 00000 condition) produces a LOW output.

3.6 (a)



(b) X = constant LOW.

(c)



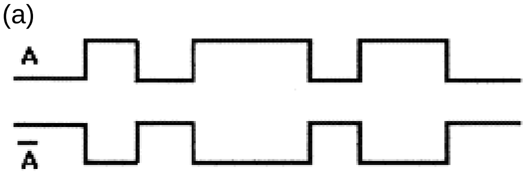
3.7 Change the OR gate to an AND gate.

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

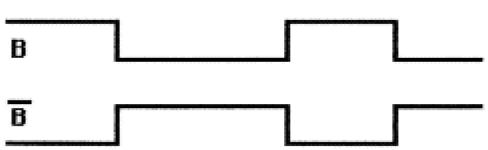
3.9 A logic HIGH and a logic LOW applied to the inputs of the unknown 2-input gate would tell us what type of gate it is. If the resulting output logic level is HIGH, then the gate is an OR gate. If the resulting output logic level is LOW, then the gate is an AND gate.

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

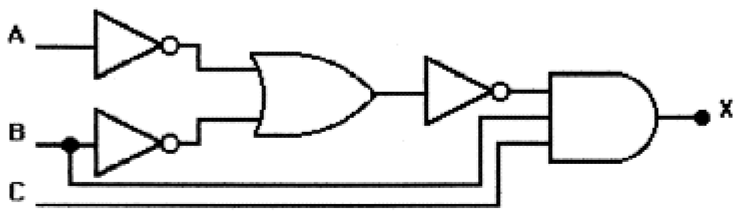
3.11 (a)



(b)



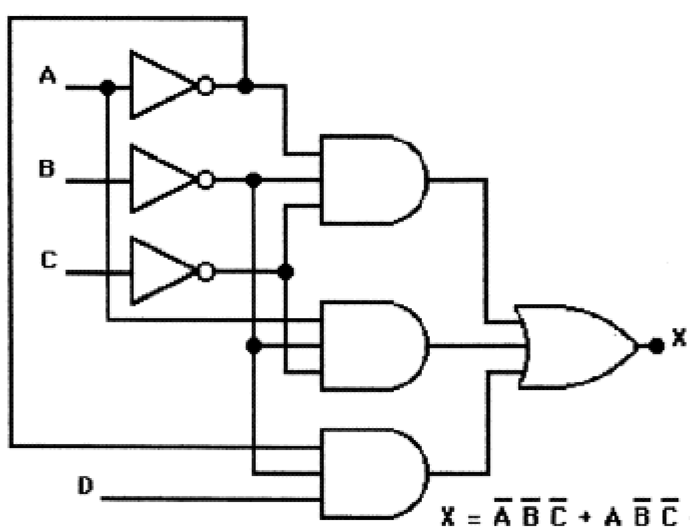
3.12 (a)



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

$$X = \left[\overline{\overline{A} + \overline{B}}\right] \cdot BC$$

(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	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$$X = \overline{A}\,\overline{B}\,\overline{C} + A\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,D$$

3.13

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

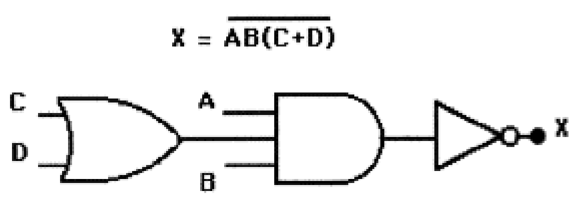
E	*D*	*C*	*B*	*A*	*AB*	*(AB)+C*	*[(AB)+C]'*	*D[(AB)+C]'*	*D[(AB)+C]'+E*
0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	0	1	0	0	0
0	0	1	0	1	0	1	0	0	0
0	0	1	1	0	0	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	1	0	0	0	0	0	1	1	1
0	1	0	0	1	0	0	1	1	1
0	1	0	1	0	0	0	1	1	1
0	1	0	1	1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

E	*D*	*C*	*B*	*A*	*AB*	*(AB)+C*	*[(AB)+C]'*	*D[(AB)+C]'*	*D[(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

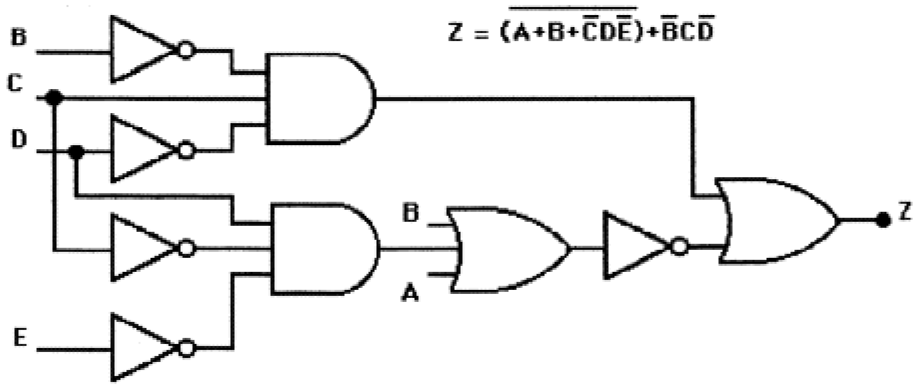
3.15

A	*B*	*C*	*D*	*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

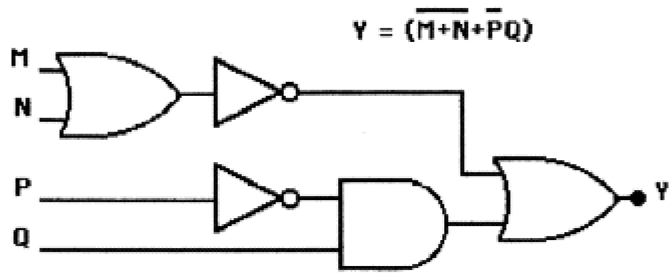
3.16 (a)



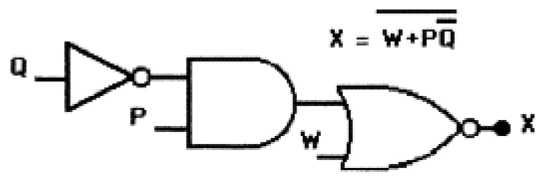
(b)



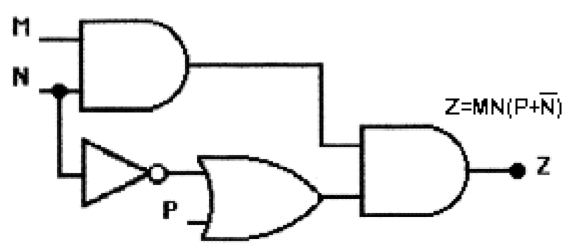
(c)



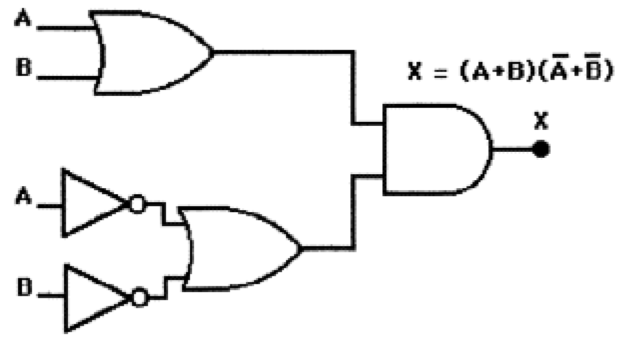
(d)



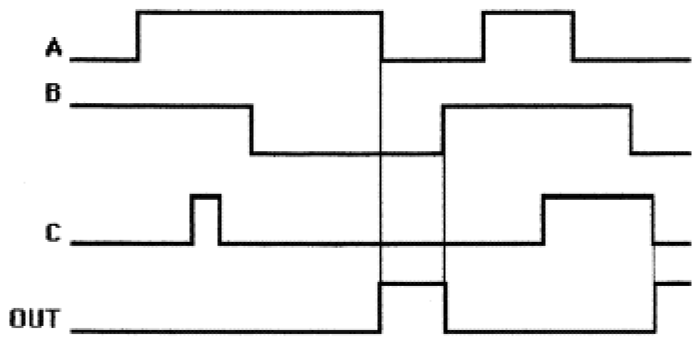
(e)



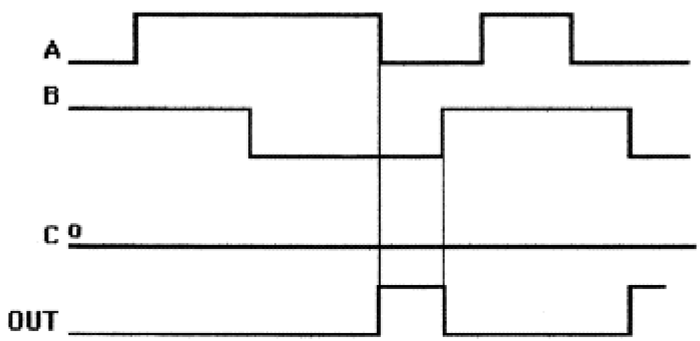
(f)



3.17 (a) OUT = 1 only when <u>all</u> inputs are = 0.

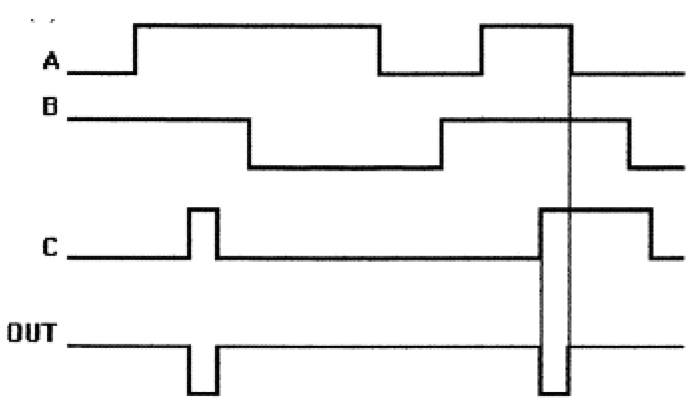


(b) With C = 0

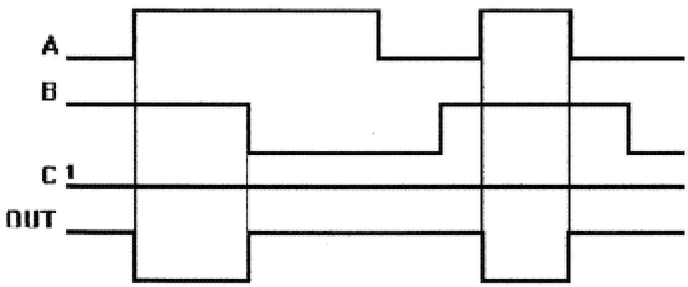


(c) With C = 1, OUT = 0 at all times.

3.18 (a) OUT = 0 only when <u>all</u> inputs are = 1.



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

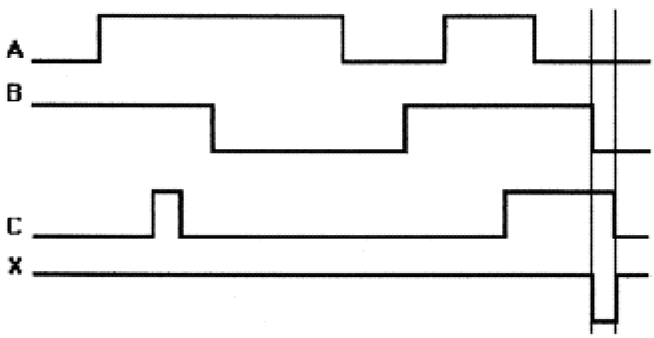


3.19

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

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

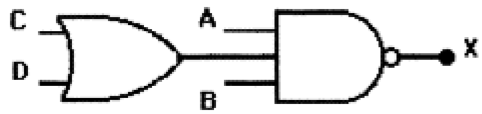
A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



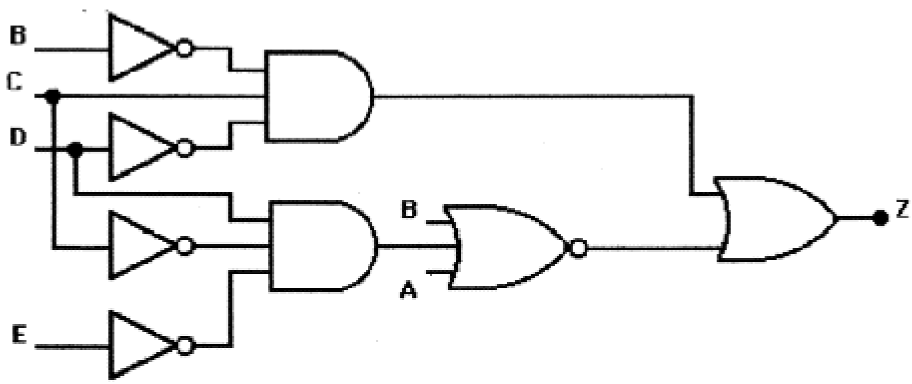
3.20

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*

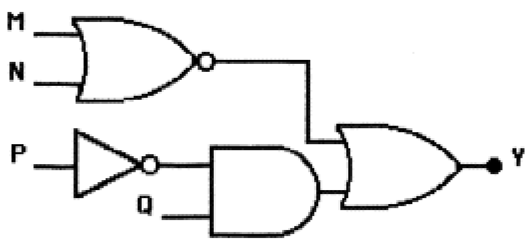
3.21 (a)



(b)



(c)



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

X=1	{ 1 + 0 1 = 1 }	**X=0**	{ 0 + 1 0 = 0 + 0 }
Y=1	{ 1 + 0 = 1 }	**Y=0**	{ 0 + 0 = 0 }
	{ **1=1** }		{ **0=0** }
X=1	{ 1 + 0 0 = 1 + 0 }	**X=0**	{ 0 + 1 1 = 0 + 1 }
Y=0	{ 1 + 0 = 1 }	**Y=1**	{ 0 + 1 = 1 }
	{ **1=1** }		{ **1=1** }

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

X=1	{ 0 + 1 1 = 1 }	**X=0**	{ 1 + 0 0 = 1 + 0 }
Y=1	{ 0 + 1 = 1 }	**Y=0**	{ 1 + 0 = 1 }
	{ **1=1** }		{ **1=1** }
X=1	{ 0 + 1 0 = 0 + 0 }	**X=0**	{ 1 + 0 1 = 1 + 0 }
Y=0	{ 0 + 0 = 0 }	**Y=1**	{ 1 + 0 = 1 }
	{ **0=0** }		{ **1=1** }

3.23

(a) $A+1=1$ (b) $A \cdot A = A$ (c) $B \cdot \overline{B} = 0$ (d) $C + C = C$ (e) $X \cdot 0 = 0$ (f) $D \cdot 1 = D$

(g) $D + 0 = D$ (h) $C + \overline{C} = 1$ (i) $G + GF = G$ (j) $Y + \overline{W}Y = Y$

3.24

(a)

$$X = (M+N)(\overline{M}+P)(\overline{N}+\overline{P})$$
$$X = (M\overline{M} + MP + N\overline{M} + NP)(\overline{N}+\overline{P})$$
$$X = (M\overline{M}\overline{N} + M\overline{M}\overline{P} + MP\overline{N} + MP\overline{P} + N\overline{M}\overline{N} + N\overline{M}\overline{P} + NP\overline{N} + NP\overline{P})$$
$$X = (0 + 0 + MP\overline{N} + 0 + 0 + N\overline{M}\overline{P} + 0 + 0)$$
$$X = MP\overline{N} + N\overline{M}\overline{P}$$

(b)

$$Z = \overline{A}B\overline{C} + AB\overline{C} + B\overline{C}D$$
$$Z = B\overline{C}(\overline{A} + A + D)$$
$$Z = B\overline{C}(1 + D)$$
$$Z = B\overline{C}$$

3.25 $\overline{A+B} = \overline{A} \cdot \overline{B}$

A=1 **B=1**	**{** $\overline{1+1} = \overline{1} \cdot 1$ **{**	**A=0** **B=0**	**{** $\overline{0+0} = \overline{0} = 1$ **{**
A=0 **B=1**	**{** $\overline{0+1} = \overline{0} \cdot \overline{1} = 0$ **{**	**A=1** **B=0**	**{** $\overline{1+0} = \overline{1} \cdot \overline{0} = 0$ **{**

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

A=1 **B=1**	**{** $\overline{1 \cdot 1} = \overline{1} + \overline{1} = 0$ **{**	**A=0** **B=0**	**{** $\overline{0 \cdot 0} = \overline{0} = 1$ **{**
A=0 **B=1**	**{** $\overline{0 \cdot 1} = \overline{0} + \overline{1} = 1$ **{**	**A=1** **B=0**	**{** $\overline{1 \cdot 0} = \overline{1} + \overline{0} = 1$ **{**

3.26

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

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

(c) $\overline{AB\overline{CD}} = \overline{AB} + \overline{\overline{CD}} = \overline{A} + \overline{B} + CD$

(d) $\overline{A + \overline{B}} = \overline{A}\overline{\overline{B}} = \overline{A}B$

(e) $\overline{\overline{A}\overline{B}} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$

(f) $\overline{\overline{A} + \overline{C} + \overline{D}} = \overline{\overline{A}}\overline{\overline{C}}\overline{\overline{D}} = ACD$

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

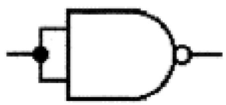
(h) $\overline{(M+\overline{N})(\overline{M}+N)} = \overline{M}N + M\overline{N}$

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

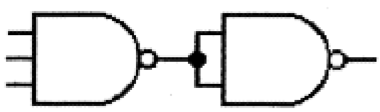
3.27

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

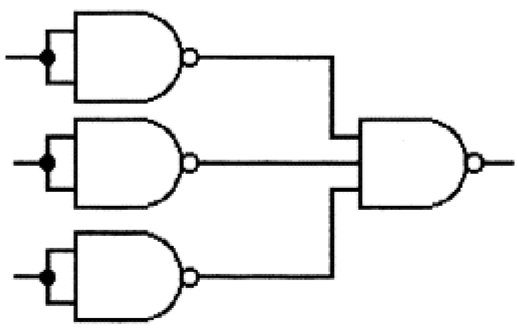
3.28 Change each inverter to:



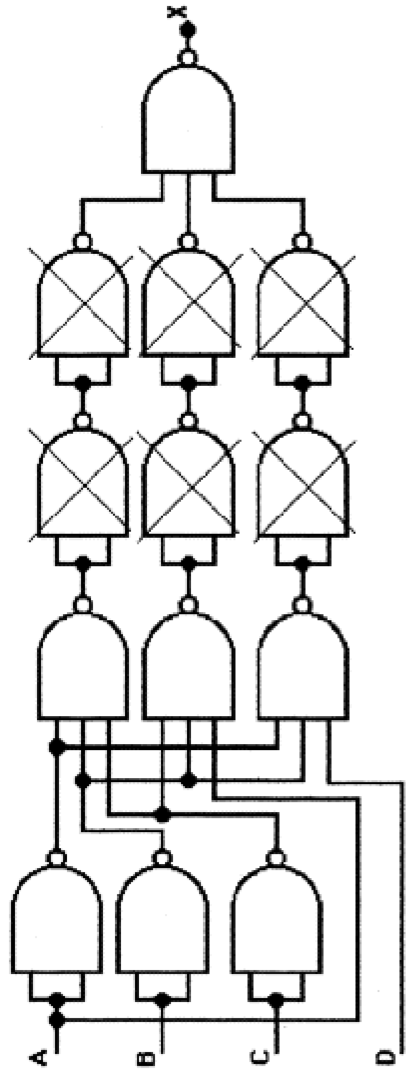
Change each AND to:



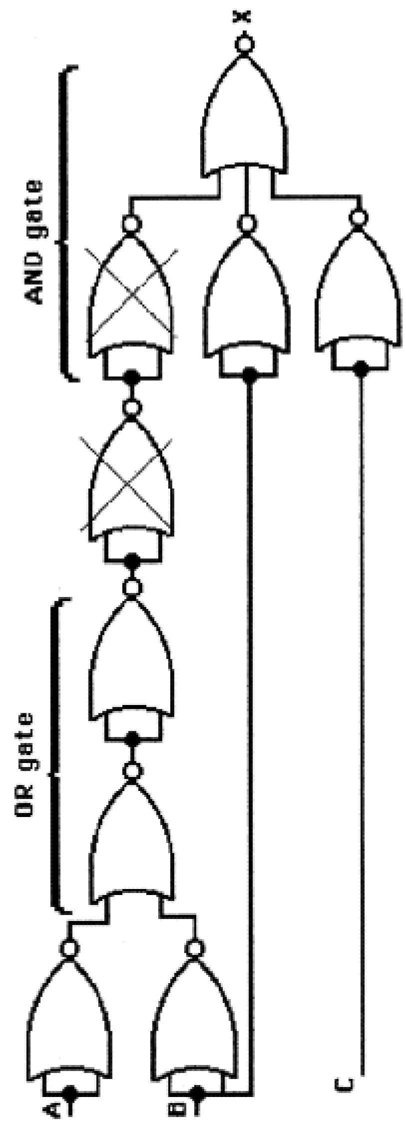
Change each OR to:



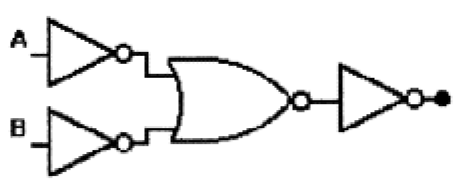
By canceling double INVERTERS, the result is: $X = \overline{A}\,\overline{B}\,\overline{C} + A\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,D$



3.29 X=ABC

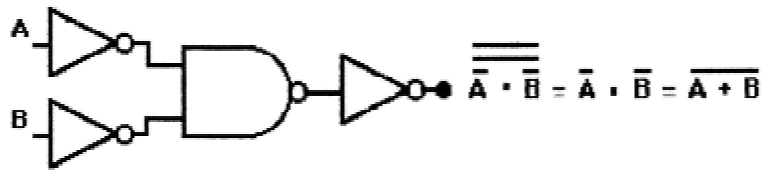


3.30



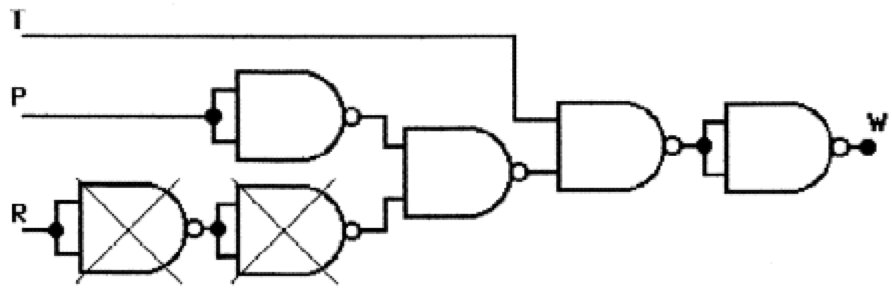
$\overline{\overline{\bar{A} + \bar{B}}} = \bar{A} + \bar{B} = \overline{AB}$

3.31



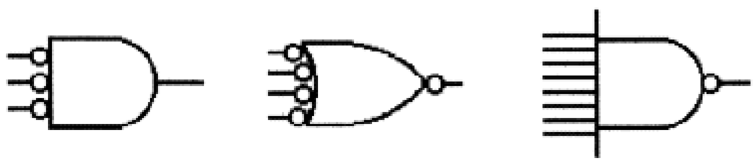
3.32 (a) The warning light W will be activated, when temperature (T) is >200°F **and** either the pressure (P) is >220 p.s.i., **or** the speed (R) is < 4800 r.p.m. In conclusion, **W=1 when T=1 and either P=1 or R=0.**

(b)

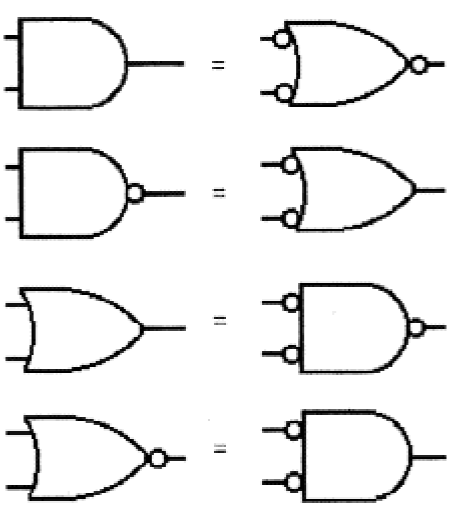


3.33

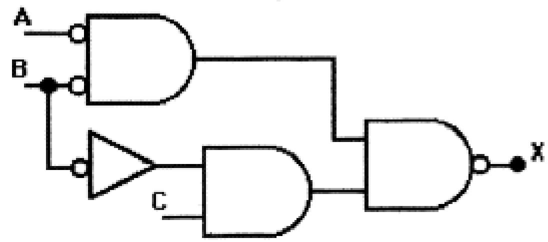
(a) NOR gate (b) AND gate (c) NAND gate



3.34



3.35 (a)

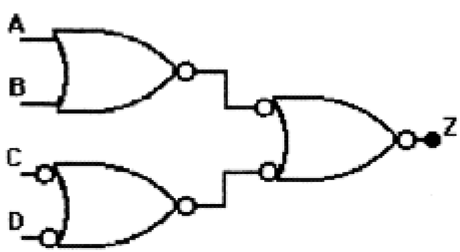


(b)

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.

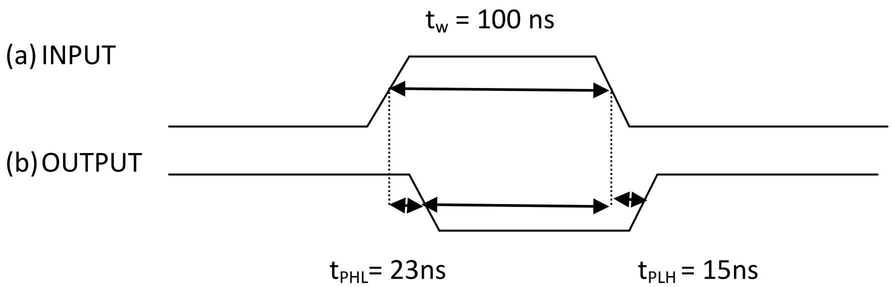
3.40

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 $\overline{\text{LIGHT}}$ = LOW when A=B=1, or when A=B=0.

B	*A*	$\overline{\text{LIGHT}}$
0	0	0
0	1	1
1	0	1
1	1	0

3.42

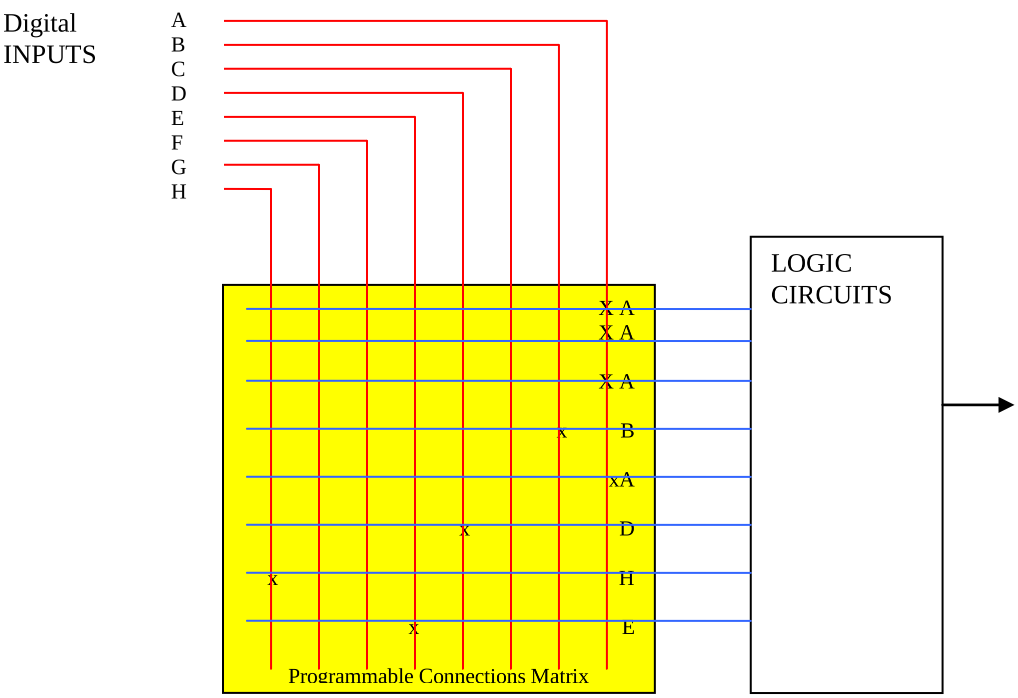


(c) t_w = 92ns

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;                    -- logic descriptions
                        y <= NOT(a AND b);
                        z <= a OR b OR c;
                END ckt;
```

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;
```

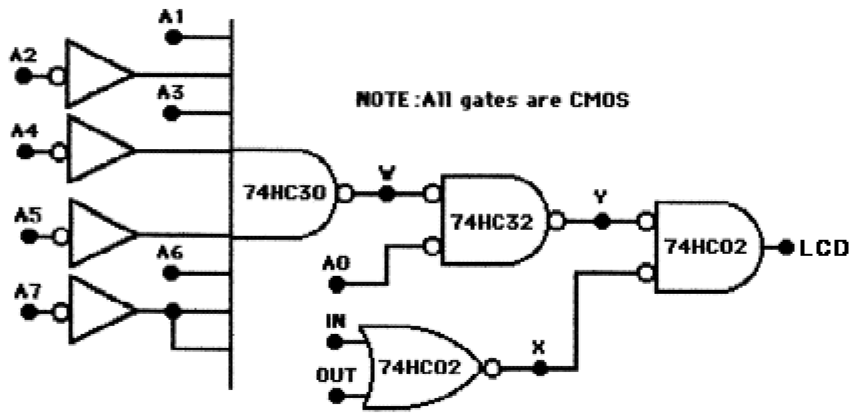
(b) AHDL

```
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;
```

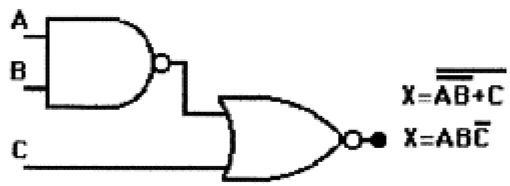
(b) VHDL

```
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;
```

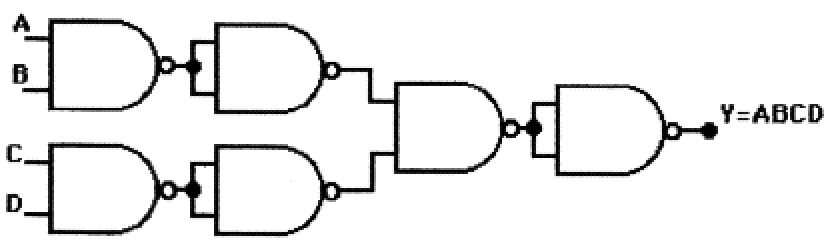
3.47


A1
A2
A3
A4
A5
A6
A7
NOTE : All gates are CMOS
74HC30
W
74HC32
Y
74HC02
LCD
A0
IN
OUT
74HC02
X

3.48


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

3.49


A
B
C
D
Y=ABCD