Digital Design and Computer Architecture 2nd Edition Harris Solutions Manual
Full Download: http://alibabadownload.com/product/digital-design-and-computer-architecture-2nd-edition-harris-solutions

CHAPTER 1

SOLUTIONS

This sample only, Download all chapters at: alibabadownload.com

CHAPTER 1

Exercise 1.1

(a) Biologists study cells at many levels. The cells are built from organelles such as the mitochondria, ribosomes, and chloroplasts. Organelles are built of macromolecules such as proteins, lipids, nucleic acids, and carbohydrates. These biochemical macromolecules are built simpler molecules such as carbon chains and amino acids. When studying at one of these levels of abstraction, biologists are usually interested in the levels above and below: what the structures at that level are used to build, and how the structures themselves are built.

(b) The fundamental building blocks of chemistry are electrons, protons, and neutrons (physicists are interested in how the protons and neutrons are built). These blocks combine to form atoms. Atoms combine to form molecules. For example, when chemists study molecules, they can abstract away the lower levels of detail so that they can describe the general properties of a molecule such as benzene without having to calculate the motion of the individual electrons in the molecule.

Exercise 1.2

(a) Automobile designers use hierarchy to construct a car from major assemblies such as the engine, body, and suspension. The assemblies are constructed from subassemblies; for example, the engine contains cylinders, fuel injectors, the ignition system, and the drive shaft. Modularity allows components to be swapped without redesigning the rest of the car; for example, the seats can be cloth, leather, or leather with a built in heater depending on the model of the vehicle, so long as they all mount to the body in the same place. Regularity involves the use of interchangeable parts and the sharing of parts between different vehicles; a 65R14 tire can be used on many different cars.

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

(b) Businesses use hierarchy in their organization chart. An employee reports to a manager, who reports to a general manager who reports to a vice president who reports to the president. Modularity includes well-defined interfaces between divisions. The salesperson who spills a coke in his laptop calls a single number for technical support and does not need to know the detailed organization of the information systems department. Regularity includes the use of standard procedures. Accountants follow a well-defined set of rules to calculate profit and loss so that the finances of each division can be combined to determine the finances of the company and so that the finances of the company can be reported to investors who can make a straightforward comparison with other companies.

Exercise 1.3

Ben can use a hierarchy to design the house. First, he can decide how many bedrooms, bathrooms, kitchens, and other rooms he would like. He can then jump up a level of hierarchy to decide the overall layout and dimensions of the house. At the top-level of the hierarchy, he material he would like to use, what kind of roof, etc. He can then jump to an even lower level of hierarchy to decide the specific layout of each room, where he would like to place the doors, windows, etc. He can use the principle of regularity in planning the framing of the house. By using the same type of material, he can scale the framing depending on the dimensions of each room. He can also use regularity to choose the same (or a small set of) doors and windows for each room. That way, when he places a new door or window he need not redesign the size, material, layout specifications from scratch. This is also an example of modularity: once he has designed the specifications for the windows in one room, for example, he need not respecify them when he uses the same windows in another room. This will save him both design time and, thus, money. He could also save by buying some items (like windows) in bulk.

Exercise 1.4

An accuracy of +/- 50 mV indicates that the signal can be resolved to 100 mV intervals. There are 50 such intervals in the range of 0-5 volts, so the signal represents $\log_2 50 = 5.64$ bits of information.

Exercise 1.5

(a) The hour hand can be resolved to 12 * 4 = 48 positions, which represents $\log_2 48 = 5.58$ bits of information. (b) Knowing whether it is before or after noon adds one more bit.

Exercise 1.6

Each digit conveys $\log_2 60 = 5.91$ bits of information. $4000_{10} = 1\ 6\ 40_{60}$ (1 in the 3600 column, 6 in the 60's column, and 40 in the 1's column).

Exercise 1.7

$2^{16} = 65{,}536$ numbers.

Exercise 1.8

$2^{32}-1 = 4{,}294{,}967{,}295$

Exercise 1.9

(a) $2^{16}-1 = 65535$; (b) $2^{15}-1 = 32767$; (c) $2^{15}-1 = 32767$

Exercise 1.10

(a) $2^{32}-1 = 4{,}294{,}967{,}295$; (b) $2^{31}-1 = 2{,}147{,}483{,}647$; (c) $2^{31}-1 = 2{,}147{,}483{,}647$

Exercise 1.11

(a) 0; (b) $-2^{15} = -32768$; (c) $-(2^{15}-1) = -32767$

Exercise 1.12

(a) 0; (b) $-2^{31} = -2{,}147{,}483{,}648$; (c) $-(2^{31}-1) = -2{,}147{,}483{,}647$;

Exercise 1.13

(a) 10; (b) 54; (c) 240; (d) 2215

Exercise 1.14

(a) 14; (b) 36; (c) 215; (d) 15,012

Exercise 1.15

(a) A; (b) 36; (c) F0; (d) 8A7

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

Exercise 1.16

(a) E; (b) 24; (c) D7; (d) 3AA4

Exercise 1.17

(a) 165; (b) 59; (c) 65535; (d) 3489660928

Exercise 1.18

(a) 78; (b) 124; (c) 60,730; (d) 1,077,915, 649

Exercise 1.19

(a) 10100101; (b) 00111011; (c) 1111111111111111;
(d) 11010000000000000000000000000000

Exercise 1.20

(a) 1001110; (b) 1111100; (c) 1110110100111010; (d) 100 0000 0011 1111 1011 0000 0000 0001

Exercise 1.21

(a) -6; (b) -10; (c) 112; (d) -97

Exercise 1.22

(a) -2 (-8+4+2 = -2 or magnitude = 0001+1 = 0010: thus, -2); (b) -29 (-32 + 2 + 1 = -29 or magnitude = 011100+1 = 011101: thus, -29); (c) 78; (d) -75

Exercise 1.23

(a) -2; (b) -22; (c) 112; (d) -31

Exercise 1.24

(a) -6; (b) -3; (c) 78; (d) -53

Exercise 1.25

(a) 101010; (b) 111111; (c) 11100101; (d) 1101001101

Exercise 1.26

(a) 1110; (b) 110100; (c) 101010011; (d) 1011000111

Exercise 1.27

(a) 2A; (b) 3F; (c) E5; (d) 34D

Exercise 1.28

(a) E; (b) 34; (c) 153; (d) 2C7;

Exercise 1.29

(a) 00101010; (b) 11000001; (c) 01111100; (d) 10000000; (e) overflow

Exercise 1.30

(a) 00011000; (b) 11000101; (c) overflow; (d) overflow; (e) 01111111\

Exercise 1.31

00101010; (b) 10111111; (c) 01111100; (d) overflow; (e) overflow

Exercise 1.32

(a) 00011000; (b) 10111011; (c) overflow; (d) overflow; (e) 01111111

Exercise 1.33

(a) 00000101; (b) 11111010

Exercise 1.34

(a) 00000111; (b) 11111001

Exercise 1.35

(a) 00000101; (b) 00001010

Exercise 1.36

(a) 00000111; (b) 00001001

Exercise 1.37

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

(a) 52; (b) 77; (c) 345; (d) 1515

Exercise 1.38

(a) 0o16; (b) 0o64; (c) 0o339; (d) 0o1307

Exercise 1.39

(a) 100010_2, 22_{16}, 34_{10}; (b) 110011_2, 33_{16}, 51_{10}; (c) 010101101_2, AD_{16}, 173_{10}; (d) 011000100111_2, 627_{16}, 1575_{10}

Exercise 1.40

(a) 0b10011; 0x13; 19; (b) 0b100101; 0x25; 37; (c) 0b11111001; 0xF9; 249; (d) 0b10101110000; 0x570; 1392

Exercise 1.41

15 greater than 0, 16 less than 0; 15 greater and 15 less for sign/magnitude

Exercise 1.42

(26-1) are greater than 0; 26 are less than 0. For sign/magnitude numbers, (26-1) are still greater than 0, but (26-1) are less than 0.

Exercise 1.43

4, 8

Exercise 1.44

8

Exercise 1.45

5,760,000

Exercise 1.46

(5 × 109 bits/second)(60 seconds/minute)(1 byte/8 bits) = 3.75 × 1010 bytes

Exercise 1.47

46.566 gigabytes

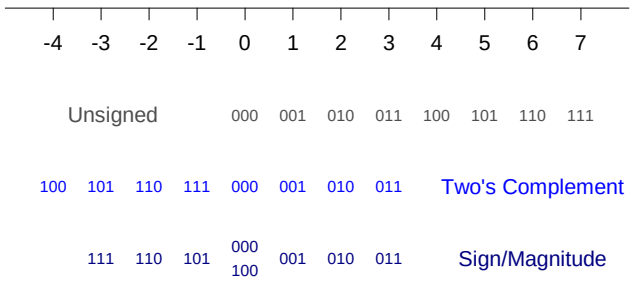
Exercise 1.48

2 billion

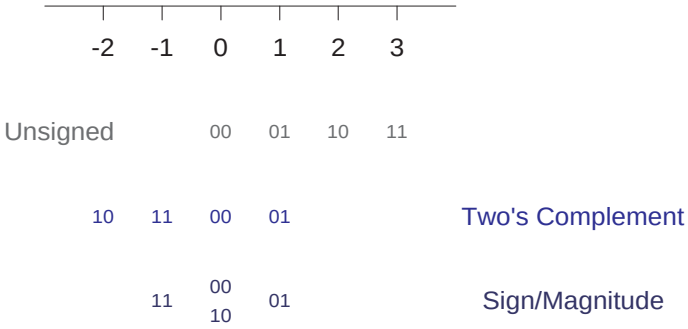
Exercise 1.49

128 kbits

Exercise 1.50



Exercise 1.51



Exercise 1.52

(a) 1101; (b) 11000 (overflows)

Exercise 1.53

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

(a) 11011101; (b) 110001000 (overflows)

Exercise 1.54

(a) 11012, no overflow; (b) 10002, no overflow

Exercise 1.55

(a) 11011101; (b) 110001000

Exercise 1.56

(a) 010000 + 001001 = 011001;
(b) 011011 + 011111 = 111010 (overflow);
(c) 111100 + 010011 = 001111;
(d) 000011 + 100000 = 100011;
(e) 110000 + 110111 = 100111;
(f) 100101 + 100001 = 000110 (overflow)

Exercise 1.57

(a) 000111 + 001101 = 010100
(b) 010001 + 011001 = 101010, overflow
(c) 100110 + 001000 = 101110
(d) 011111 + 110010 = 010001
(e) 101101 + 101010 = 010111, overflow
(f) 111110 + 100011 = 100001

Exercise 1.58

(a) 10; (b) 3B; (c) E9; (d) 13C (overflow)

Exercise 1.59

(a) 0x2A; (b) 0x9F; (c) 0xFE; (d) 0x66, overflow

Exercise 1.60

(a) 01001 - 00111 = 00010; (b) 01100 - 01111 = 11101; (c) 11010 - 01011 = 01111; (d) 00100 - 11000 = 01100

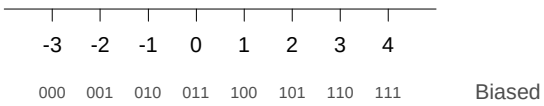
Exercise 1.61

(a) 010010 + 110100 = 000110; (b) 011110 + 110111 = 010101; (c) 100100 + 111101 = 100001; (d) 110000 + 101011 = 011011, overflow

Exercise 1.62

(a) 3; (b) 01111111; (c) $00000000_2 = -127_{10}$; $11111111_2 = 128_{10}$

Exercise 1.63



Exercise 1.64

(a) 001010001001; (b) 951; (c) 1000101; (d) each 4-bit group represents one decimal digit, so conversion between binary and decimal is easy. BCD can also be used to represent decimal fractions exactly.

Exercise 1.65

(a) 0011 0111 0001
(b) 187
(c) 95 = 1011111
(d) Addition of BCD numbers doesn't work directly. Also, the representation doesn't maximize the amount of information that can be stored; for example 2 BCD digits requires 8 bits and can store up to 100 values (0-99) - unsigned 8-bit binary can store 28 (256) values.

Exercise 1.66

Three on each hand, so that they count in base six.

Exercise 1.67

Both of them are full of it. $42_{10} = 101010_2$, which has 3 1's in its representation.

Exercise 1.68

Both are right.

Exercise 1.69

```
#include <stdio.h>
```

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

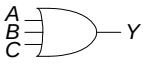

```
void main(void)
{
   char bin[80];
   int i = 0, dec = 0;

   printf("Enter binary number: ");
   scanf("%s", bin);

   while (bin[i] != 0) {
      if (bin[i] == '0') dec = dec * 2;
      else if (bin[i] == '1') dec = dec * 2 + 1;
      else printf("Bad character %c in the number.\n", bin[i]);
      i = i + 1;
   }
   printf("The decimal equivalent is %d\n", dec);
}
```

Exercise 1.70

```
/* This program works for numbers that don't overflow the
   range of an integer. */

#include <stdio.h>

void main(void)
{
   int b1, b2, digits1 = 0, digits2 = 0;
   char num1[80], num2[80], tmp, c;
   int digit, num = 0, j;

   printf ("Enter base #1: "); scanf("%d", &b1);
   printf ("Enter base #2: "); scanf("%d", &b2);
   printf ("Enter number in base %d ", b1); scanf("%s", num1);

   while (num1[digits1] != 0) {
      c = num1[digits1++];
      if (c >= 'a' && c <= 'z') c = c + 'A' - 'a';
      if (c >= '0' && c <= '9') digit = c - '0';
      else if (c >= 'A' && c <= 'F') digit = c - 'A' + 10;
      else printf("Illegal character %c\n", c);
      if (digit >= b1) printf("Illegal digit %c\n", c);
      num = num * b1 + digit;

   }
   while (num > 0) {
      digit = num % b2;
      num = num / b2;
      num2[digits2++] = digit < 10 ? digit + '0' : digit + 'A' -
10;
   }
   num2[digits2] = 0;

   for (j = 0; j < digits2/2; j++) { // reverse order of digits
      tmp = num2[j];
      num2[j] = num2[digits2-j-1];
      num2[digits2-j-1] = tmp;
   }

   printf("The base %d equivalent is %s\n", b2, num2);
}
```

Exercise 1.71

OR3

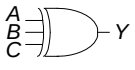


$Y = A+B+C$

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(a)

XOR3

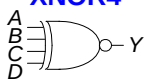


$Y = A \oplus B \oplus C$

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(b)

XNOR4



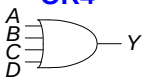
$Y = \overline{A \oplus B \oplus C \oplus D}$

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

(c)

Exercise 1.72

OR4

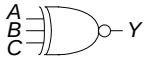


$Y = A+B+C+D$

(a)

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

XNOR3

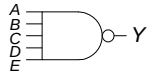


$Y = \overline{A \oplus B \oplus C}$

(b)

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

NAND5



$Y = \overline{ABCDE}$

(c)

A	B	C	D	E	Y
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	1
0	1	1	0	0	1
0	1	1	0	1	1
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	0

Exercise 1.73

A	*B*	*C*	*Y*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Exercise 1.74

A	*B*	*C*	*Y*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Exercise 1.75

A	*B*	*C*	*Y*
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Exercise 1.76

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	0

Zero

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

A NOR B

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	0

$\overline{A}B$

A	B	Y
0	0	1
0	1	1
1	0	0
1	1	0

NOT A

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	0

$A\overline{B}$

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	0

NOT B

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

XOR

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NAND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

XNOR

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

B

A	B	Y
0	0	1
0	1	1
1	0	0
1	1	1

$\overline{A} + B$

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	1

A

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	1

$A + \overline{B}$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	1

One

Exercise 1.77

2^{2^N}

Exercise 1.78

$V_{IL} = 2.5$; $V_{IH} = 3$; $V_{OL} = 1.5$; $V_{OH} = 4$; $NM_L = 1$; $NM_H = 1$

Exercise 1.79

No, there is no legal set of logic levels. The slope of the transfer characteristic never is better than -1, so the system never has any gain to compensate for noise.

Exercise 1.80

$V_{IL} = 2$; $V_{IH} = 4$; $V_{OL} = 1$; $V_{OH} = 4.5$; $NM_L = 1$; $NM_H = 0.5$

Exercise 1.81

The circuit functions as a buffer with logic levels V_{IL} = 1.5; V_{IH} = 1.8; V_{OL} = 1.2; V_{OH} = 3.0. It can receive inputs from LVCMOS and LVTTL gates because their output logic levels are compatible with this gate's input levels. However, it cannot drive LVCMOS or LVTTL gates because the 1.2 V_{OL} exceeds the V_{IL} of LVCMOS and LVTTL.

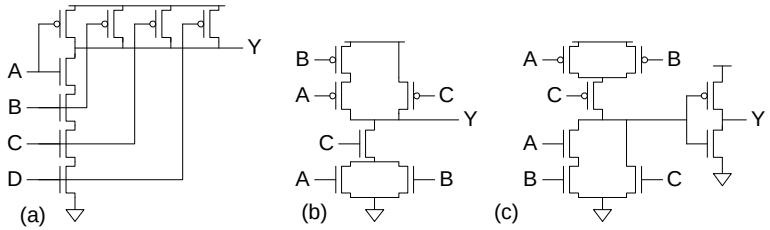
Exercise 1.82

(a) AND gate; (b) V_{IL} = 1.5; V_{IH} = 2.25; V_{OL} = 0; V_{OH} = 3

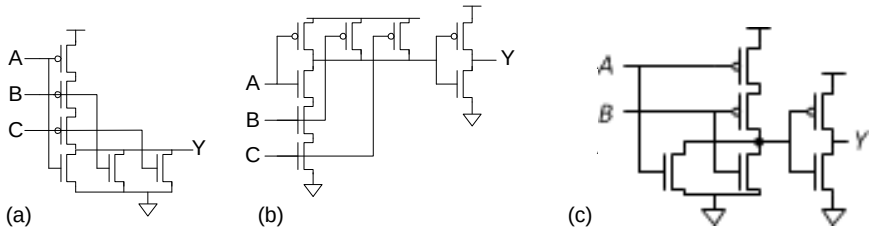
Exercise 1.83

(a) XOR gate; (b) V_{IL} = 1.25; V_{IH} = 2; V_{OL} = 0; V_{OH} = 3

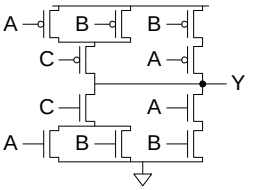
Exercise 1.84



Exercise 1.85



Exercise 1.86



Exercise 1.87

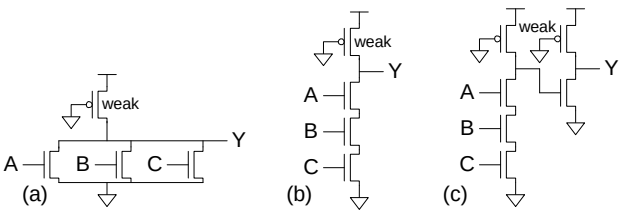
XOR

A	*B*	*Y*
0	0	0
0	1	1
1	0	1
1	1	0

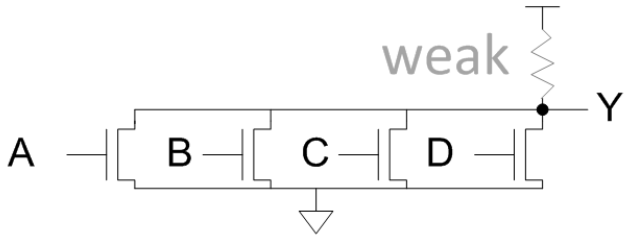
Exercise 1.88

A	*B*	*C*	*Y*
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

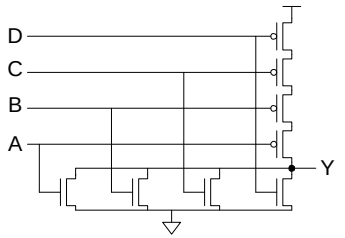
Exercise 1.89



Exercise 1.90



Question 1.1



Question 1.2

4 times. Place 22 coins on one side and 22 on the other. If one side rises, the fake is on that side. Otherwise, the fake is among the 20 remaining. From the group containing the fake, place 8 on one side and 8 on the other. Again, identify which group contains the fake. From that group, place 3 on one side and 3 on the other. Again, identify which group contains the fake. Finally, place 1 coin on each side. Now the fake coin is apparent.

Question 1.3

17 minutes: (1) designer and freshman cross (2 minutes); (2) freshman returns (1 minute); (3) professor and TA cross (10 minutes); (4) designer returns (2 minutes); (5) designer and freshman cross (2 minutes).

CHAPTER 2

Exercise 2.1

(a) $Y = \bar{A}\bar{B} + A\bar{B} + AB$

(b) $Y = \bar{A}\bar{B}\bar{C} + ABC$

(c) $Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + ABC$

(d)

$$Y = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + ABC\bar{D}$$

(e)

$$Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + AB\bar{C}\bar{D} + ABCD$$

Exercise 2.2

(a) $Y = \bar{A}B + A\bar{B} + AB$

(b) $Y = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$

(c) $Y = \bar{A}\bar{B}C + AB\bar{C} + ABC$

(d) $Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$

(e) $Y = \bar{A}\bar{B}CD + \bar{A}BC\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD$

Exercise 2.3

(a) $Y = (A + \bar{B})$

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

(b)

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

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

(d)

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

(e)

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

Exercise 2.4

(a) $Y = A + B$

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

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

(d)

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

(e)

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

Exercise 2.5

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

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

(c) $Y = \overline{A}\,\overline{C} + A\overline{B} + AC$

(d) $Y = \overline{A}\,\overline{B} + \overline{B}\,\overline{D} + AC\overline{D}$

(e)

$$Y = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} + AB\overline{C}\,\overline{D} + ABCD$$

This can also be expressed as:

$$Y = \overline{(A \oplus B)}\,\overline{(C \oplus D)} + (A \oplus B)(C \oplus D)$$

Exercise 2.6

(a) $Y = A + B$

(b) $Y = A\bar{C} + \bar{A}C + B\bar{C}$ or $Y = A\bar{C} + \bar{A}C + \bar{A}B$

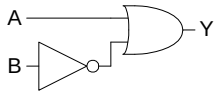
(c) $Y = AB + \bar{A}\bar{B}C$

(d) $Y = BC + \bar{B}\bar{D}$

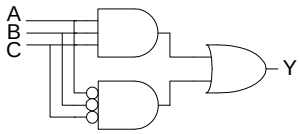
(e) $Y = A\bar{B} + \bar{A}BC + \bar{A}CD$ or $Y = A\bar{B} + \bar{A}BC + \bar{B}CD$

Exercise 2.7

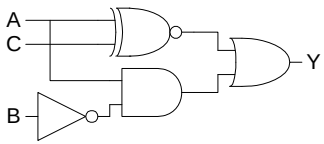
(a)



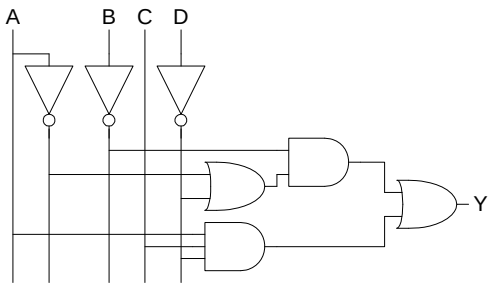
(b)



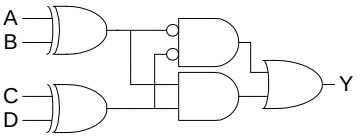
(c)



(d)

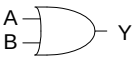


(e)

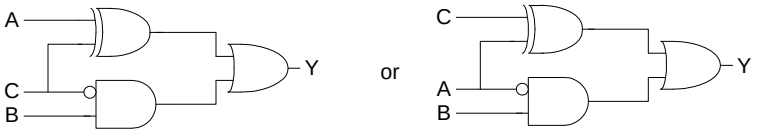


Exercise 2.8

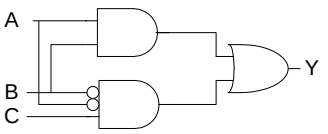
(a)



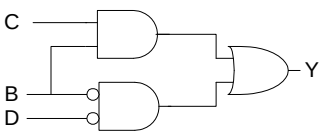
(b)



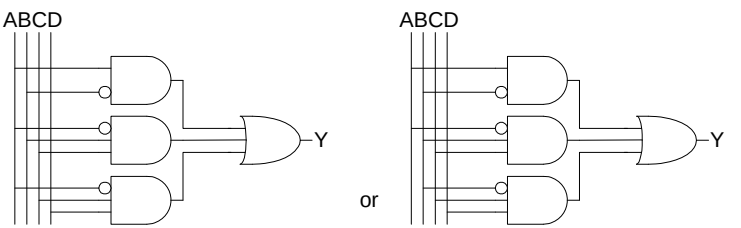
(c)



(d)

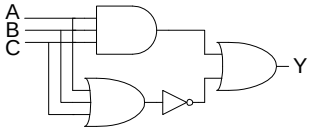


(e)

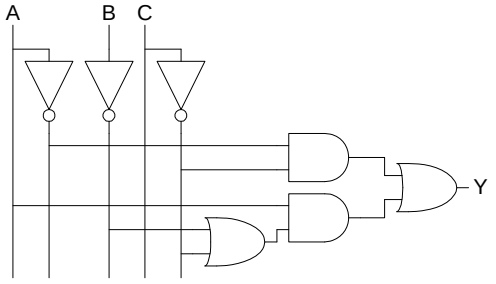


Exercise 2.9

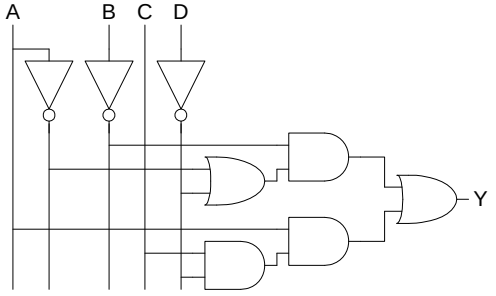
(a) Same as 2.7(a)
(b)



(c)

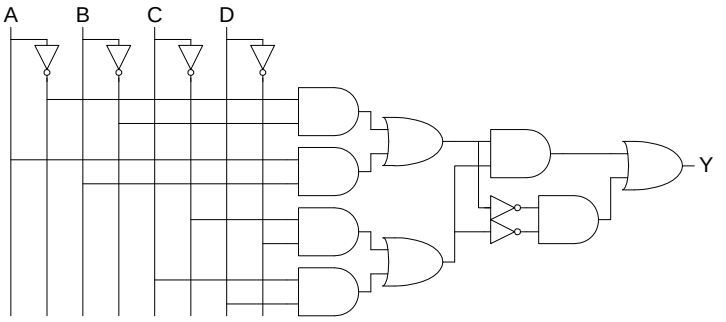


(d)

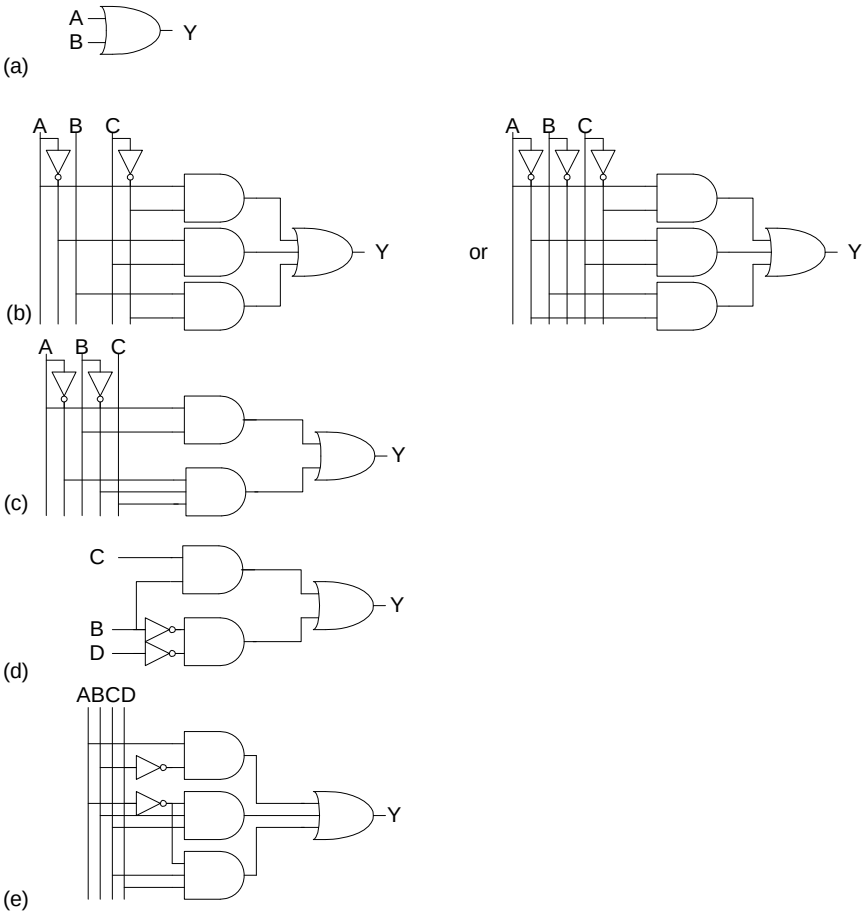


David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

(e)

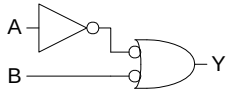


Exercise 2.10

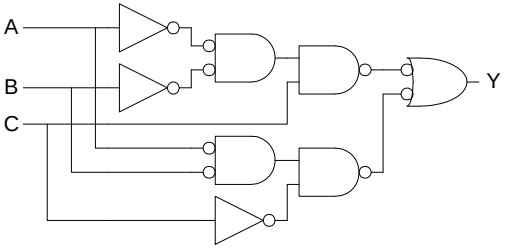


Exercise 2.11

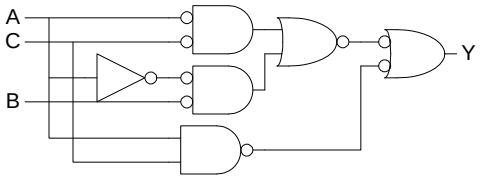
(a)



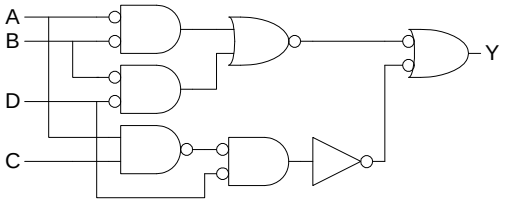
(b)



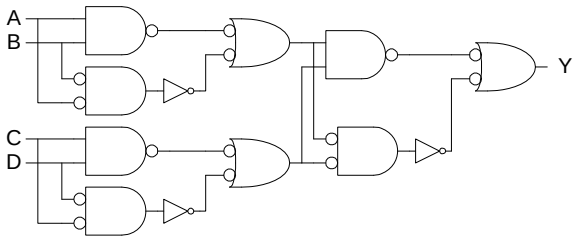
(c)



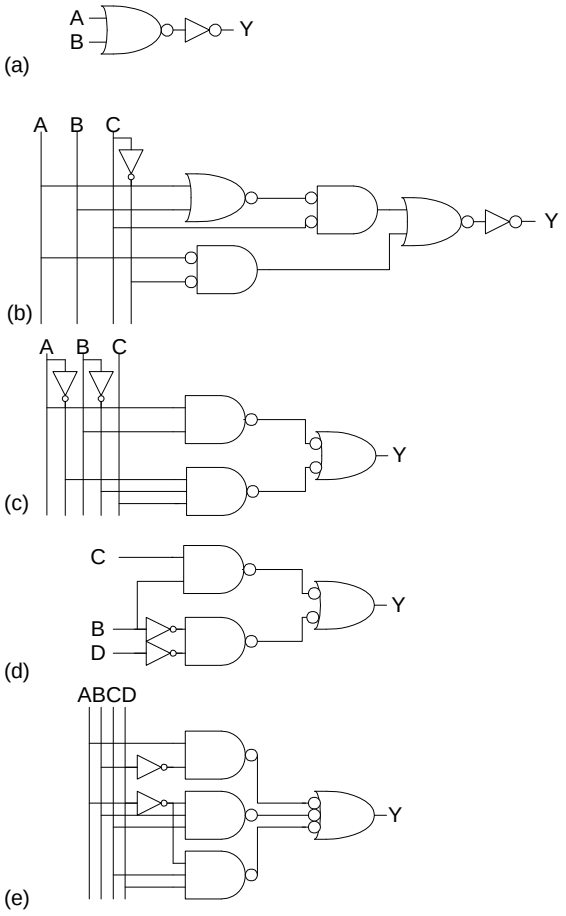
(d)



(e)



Exercise 2.12



Exercise 2.13

(a) $Y = AC + \overline{B}C$
(b) $Y = \overline{A}$
(c) $Y = \overline{A} + \overline{B}\,\overline{C} + \overline{B}\,\overline{D} + BD$

Exercise 2.14

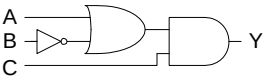
(a) $Y = \overline{A}B$
(b) $Y = \overline{A} + \overline{B} + \overline{C} = \overline{ABC}$

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

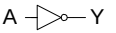
(c) $Y = A(\overline{B} + \overline{C} + \overline{D}) + \overline{B}\overline{C}\overline{D} = A\overline{BCD} + \overline{B}\overline{C}\overline{D}$

Exercise 2.15

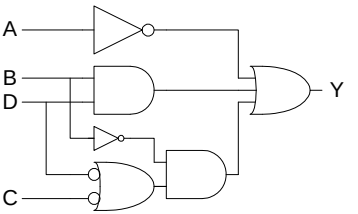
(a)



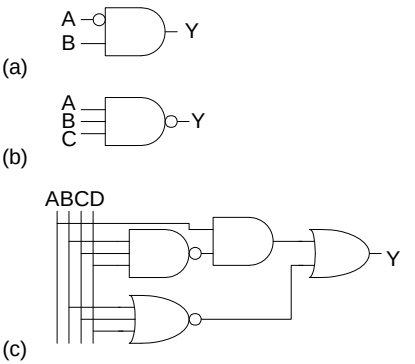
(b)



(c)



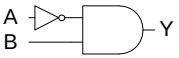
Exercise 2.16



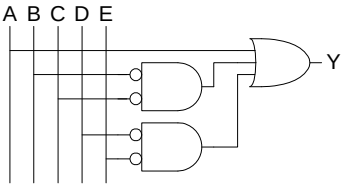
Exercise 2.17

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

(b) $Y = \overline{A}B$



(c) $Y = A + \overline{B}\,\overline{C} + \overline{D}\,\overline{E}$



Exercise 2.18

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

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

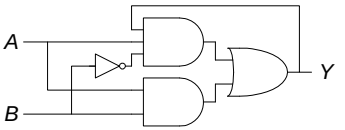
(c) $Y = B\overline{D}E + BD(\overline{A \oplus C})$

Exercise 2.19

4 gigarows = 4 x 2^{30} rows = 2^{32} rows, so the truth table has 32 inputs.

Exercise 2.20

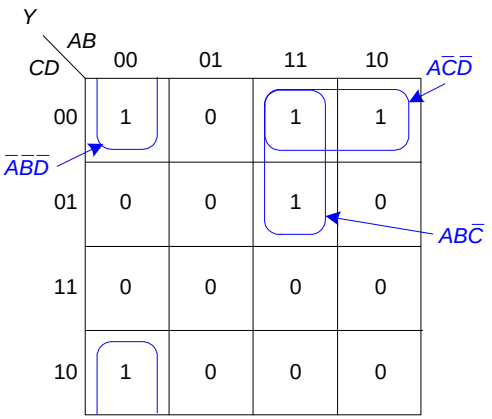
David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions



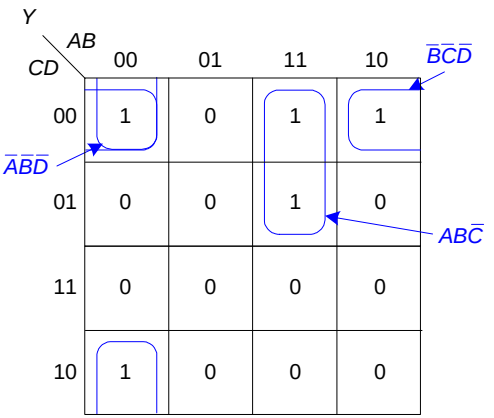
$Y = A$

Exercise 2.21

Ben is correct. For example, the following function, shown as a K-map, has two possible minimal sum-of-products expressions. Thus, although $A\bar{C}\bar{D}$ and $\bar{B}\bar{C}\bar{D}$ are both prime implicants, the minimal sum-of-products expression does not have both of them.



$Y = \bar{A}\bar{B}\bar{D} + AB\bar{C} + A\bar{C}\bar{D}$



$Y = \bar{A}\bar{B}\bar{D} + AB\bar{C} + \bar{B}\bar{C}\bar{D}$

Exercise 2.22

(a)

B	$B \bullet B$
0	0
1	1

(b)

B	C	D	$(B \bullet C) + (B \bullet D)$	$B \bullet (C + D)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

(c)

B	C	$(B \bullet C) + (B \bullet \overline{C})$
0	0	0
0	1	0
1	0	1
1	1	1

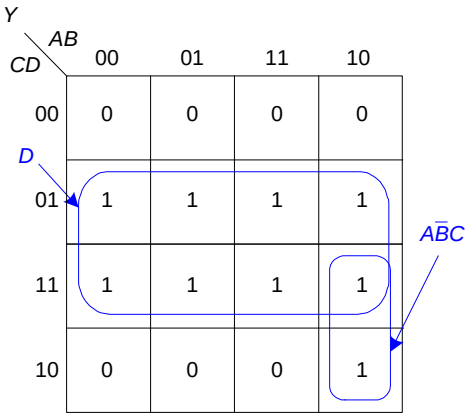
Exercise 2.23

B_2	B_1	B_0	$\overline{B_2 \bullet B_1 \bullet B_0}$	$\overline{B}_2 + \overline{B}_1 + \overline{B}_0$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

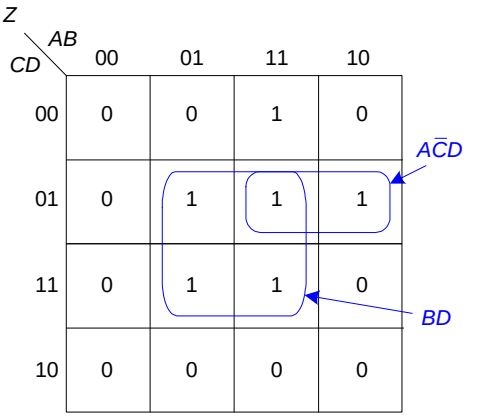
Exercise 2.24

$$Y = \overline{A}D + A\overline{B}C + A\overline{C}D + ABCD$$
$$Z = A\overline{C}D + BD$$

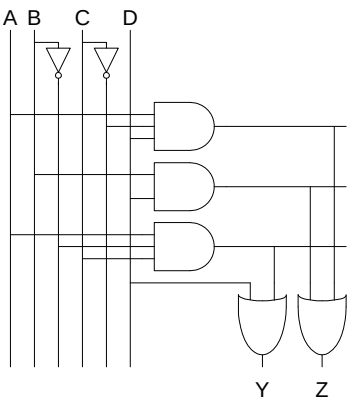
Exercise 2.25



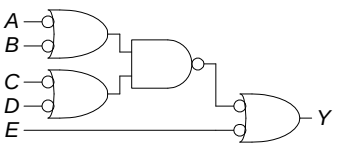
$Y = A\bar{B}C + D$



$Z = A\bar{C}D + BD$

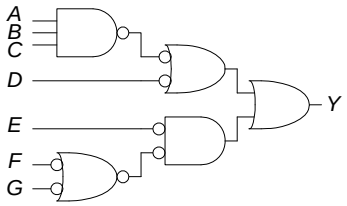


Exercise 2.26



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

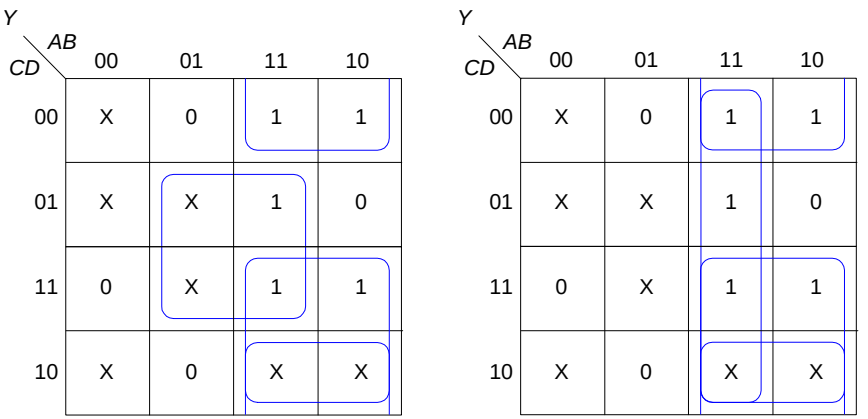
Exercise 2.27



$Y = ABC + \bar{D} + (\bar{F} + \bar{G})\bar{E}$
$= ABC + \bar{D} + \bar{E}\bar{F} + \bar{E}\bar{G}$

Exercise 2.28

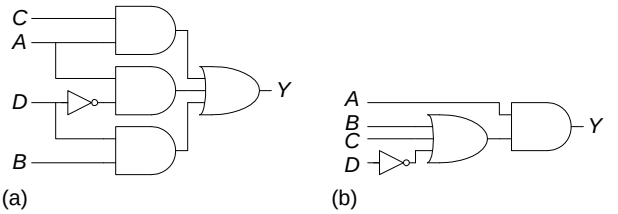
Two possible options are shown below:



Exercise 2.29

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

Two possible options are shown below:



Exercise 2.30

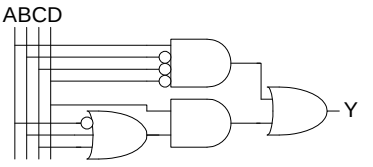
Option (a) could have a glitch when A=1, B=1, C=0, and D transitions from 1 to 0. The glitch could be removed by instead using the circuit in option (b).

Option (b) does not have a glitch. Only one path exists from any given input to the output.

Exercise 2.31

$$Y = \bar{A}D + A\bar{B}\bar{C}\bar{D} + BD + CD = A\bar{B}\bar{C}\bar{D} + D(\bar{A} + B + C)$$

Exercise 2.32



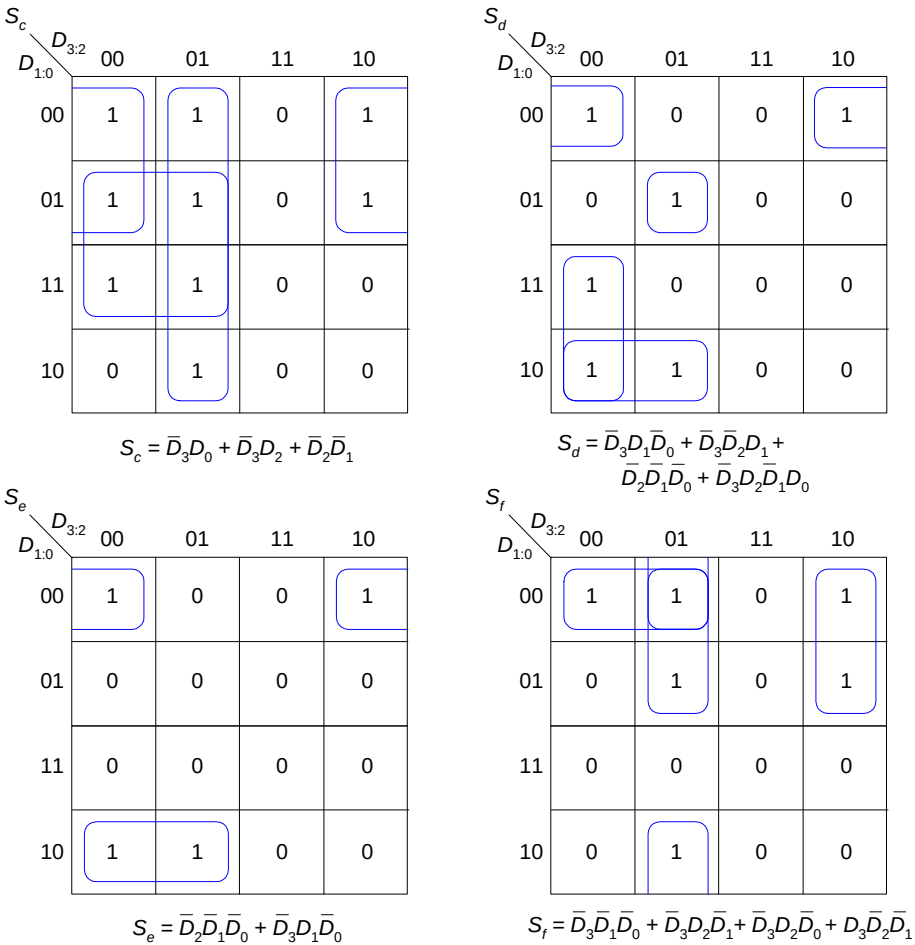
Exercise 2.33

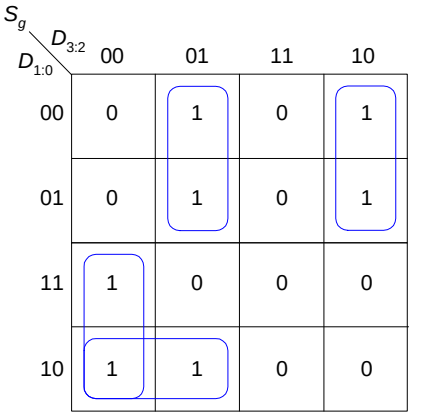
The equation can be written directly from the description:

$$E = S\bar{A} + AL + H$$

Exercise 2.34

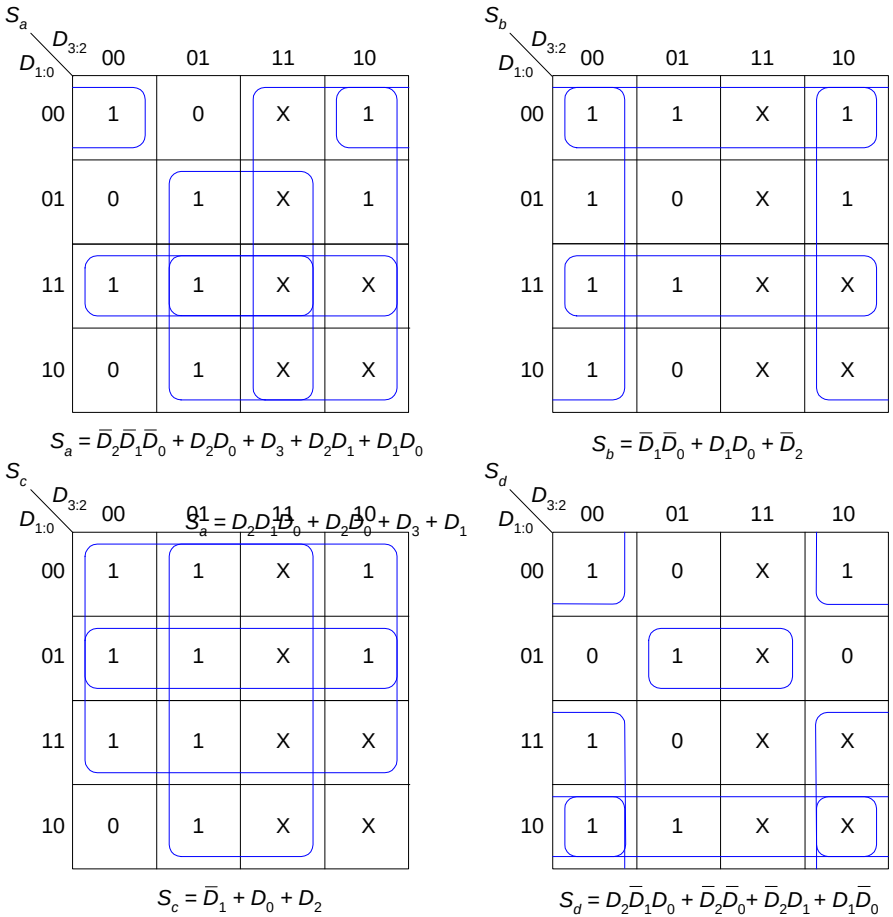
(a)



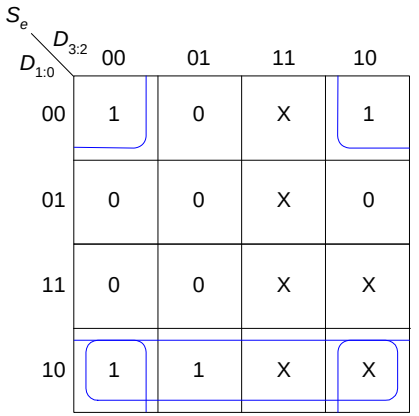


$$S_g = \bar{D}_3\bar{D}_2D_1 + \bar{D}_3D_1\bar{D}_0 + \bar{D}_3D_2\bar{D}_1 + D_3\bar{D}_2\bar{D}_1$$

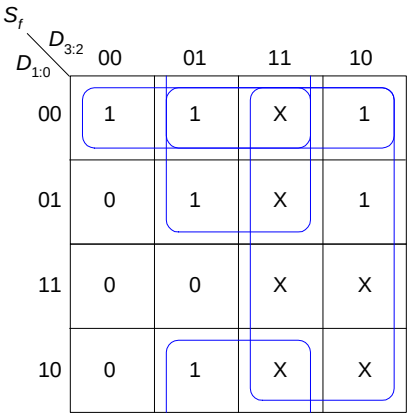
(b)



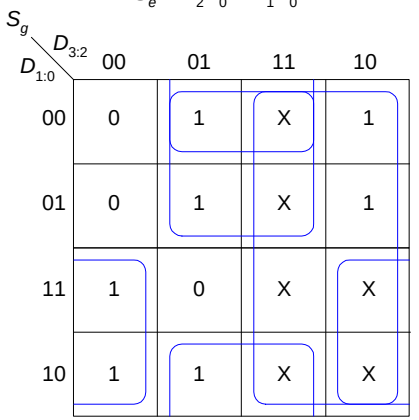
$S_a = D_2\bar{D}_1D_0 + D_2\bar{D}_0 + D_3 + D_1$



$S_e = \bar{D}_2\bar{D}_0 + D_1\bar{D}_0$

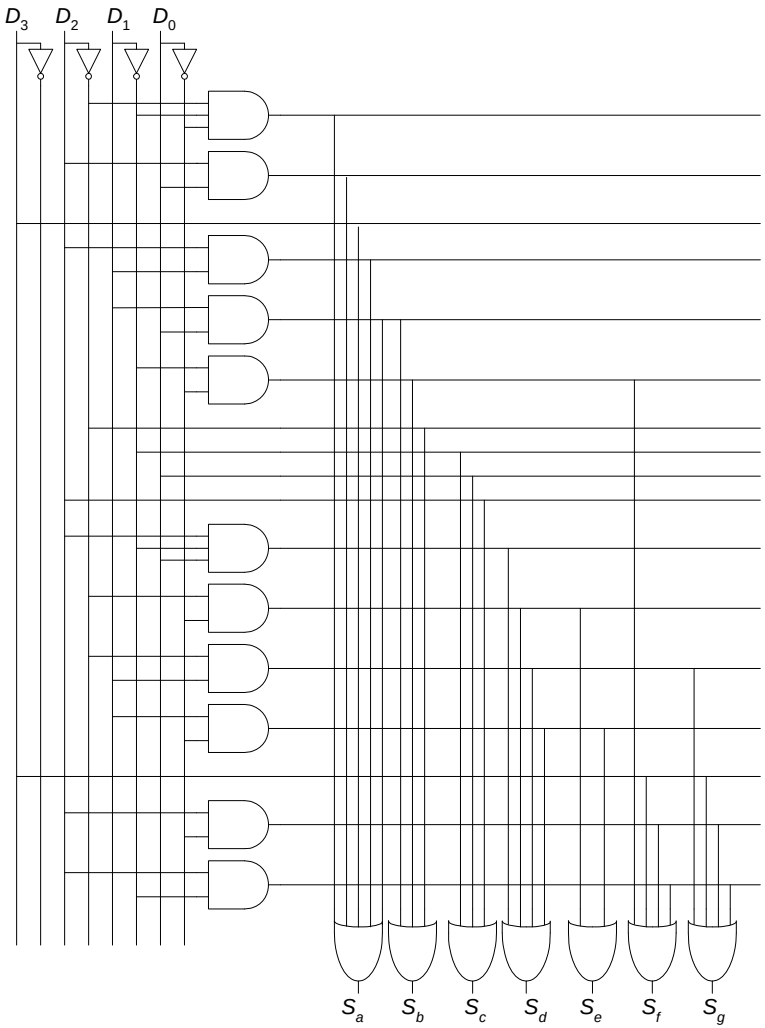


$S_f = \bar{D}_1\bar{D}_0 + D_2\bar{D}_1 + D_2\bar{D}_0 + D_3$



$S_g = \bar{D}_2D_1 + D_2\bar{D}_0 + D_2\bar{D}_1 + D_3$

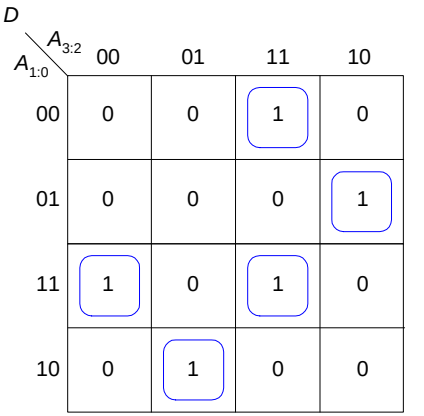
(c)



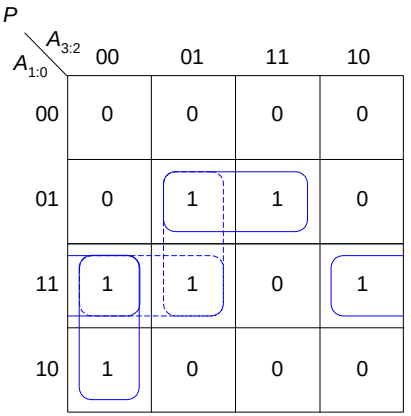
Exercise 2.35

Decimal Value	A_3	A_2	A_1	A_0	D	P
0	0	0	0	0	0	0
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	1
4	0	1	0	0	0	0
5	0	1	0	1	0	1
6	0	1	1	0	1	0
7	0	1	1	1	0	1
8	1	0	0	0	0	0
9	1	0	0	1	1	0
10	1	0	1	0	0	0
11	1	0	1	1	0	1
12	1	1	0	0	1	0
13	1	1	0	1	0	1
14	1	1	1	0	0	0
15	1	1	1	1	1	0

P has two possible minimal solutions:



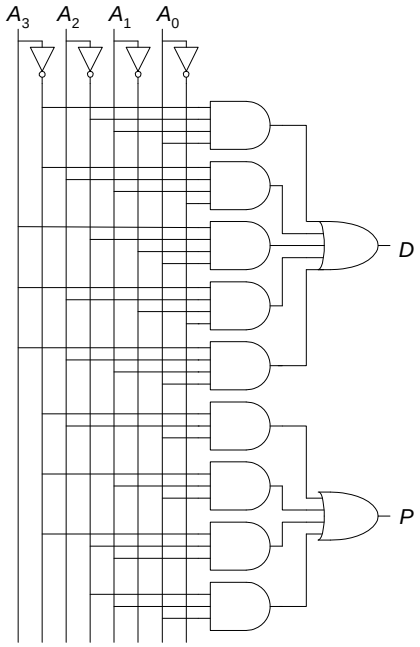
$$D = \bar{A}_3\bar{A}_2A_1A_0 + \bar{A}_3A_2A_1\bar{A}_0 + A_3\bar{A}_2\bar{A}_1A_0 + A_3A_2\bar{A}_1\bar{A}_0 + A_3A_2A_1A_0$$



$$P = \bar{A}_3A_2A_0 + \bar{A}_3A_1A_0 + \bar{A}_3\bar{A}_2A_1 + \bar{A}_2A_1A_0$$

$$P = \bar{A}_3A_1A_0 + \bar{A}_3\bar{A}_2A_1 + \bar{A}_2A_1A_0 + A_2\bar{A}_1A_0$$

Hardware implementations are below (implementing the first minimal equation given for P).



Exercise 2.36

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Y_2	Y_1	Y_0	*NONE*
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	X	0	0	1	0
0	0	0	0	0	1	X	X	0	1	0	0
0	0	0	0	1	X	X	X	0	1	1	0
0	0	0	1	X	X	X	X	1	0	0	0
0	0	1	X	X	X	X	X	1	0	1	0
0	1	X	X	X	X	X	X	1	1	0	0
1	X	X	X	X	X	X	X	1	1	1	0

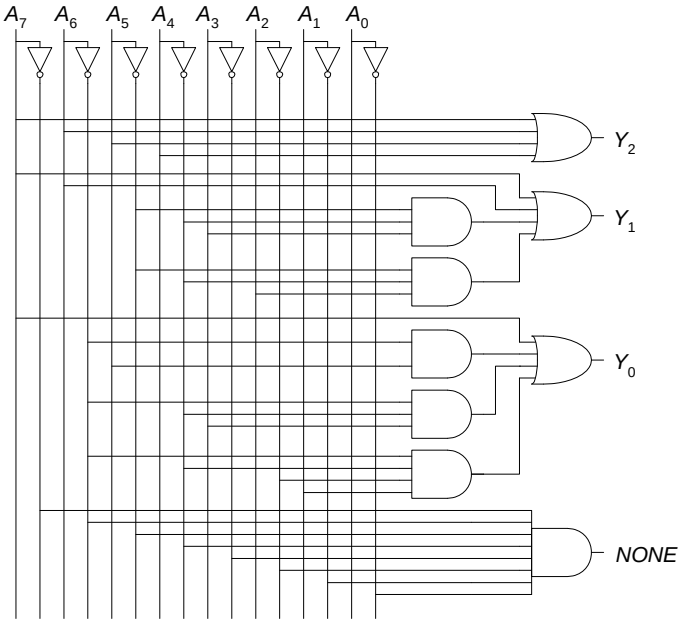
$$Y_2 = A_7 + A_6 + A_5 + A_4$$

$$Y_1 = A_7 + A_6 + \overline{A_5}\,\overline{A_4}A_3 + \overline{A_5}\,\overline{A_4}A_2$$

$$Y_0 = A_7 + \overline{A_6}A_5 + \overline{A_6}\,\overline{A_4}A_3 + \overline{A_6}\,\overline{A_4}\,\overline{A_2}A_1$$

$$NONE = \overline{A_7}\,\overline{A_6}\,\overline{A_5}\,\overline{A_4}\,\overline{A_3}\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}$$

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions



Exercise 2.37

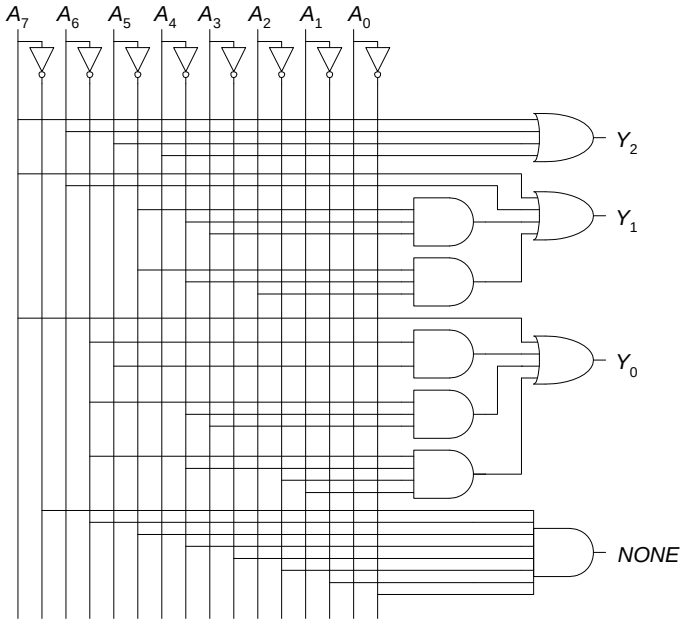
The equations and circuit for $Y_{2:0}$ is the same as in Exercise 2.25, repeated here for convenience.

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

$$Y_2 = A_7 + A_6 + A_5 + A_4$$

$$Y_1 = A_7 + A_6 + \overline{A_5}\,\overline{A_4}A_3 + \overline{A_5}\,\overline{A_4}A_2$$

$$Y_0 = A_7 + \overline{A_6}A_5 + \overline{A_6}\,\overline{A_4}A_3 + \overline{A_6}\,\overline{A_4}\,\overline{A_2}A_1$$


A_7
A_6
A_5
A_4
A_3
A_2
A_1
A_0
Y_2
Y_1
Y_0
NONE

David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

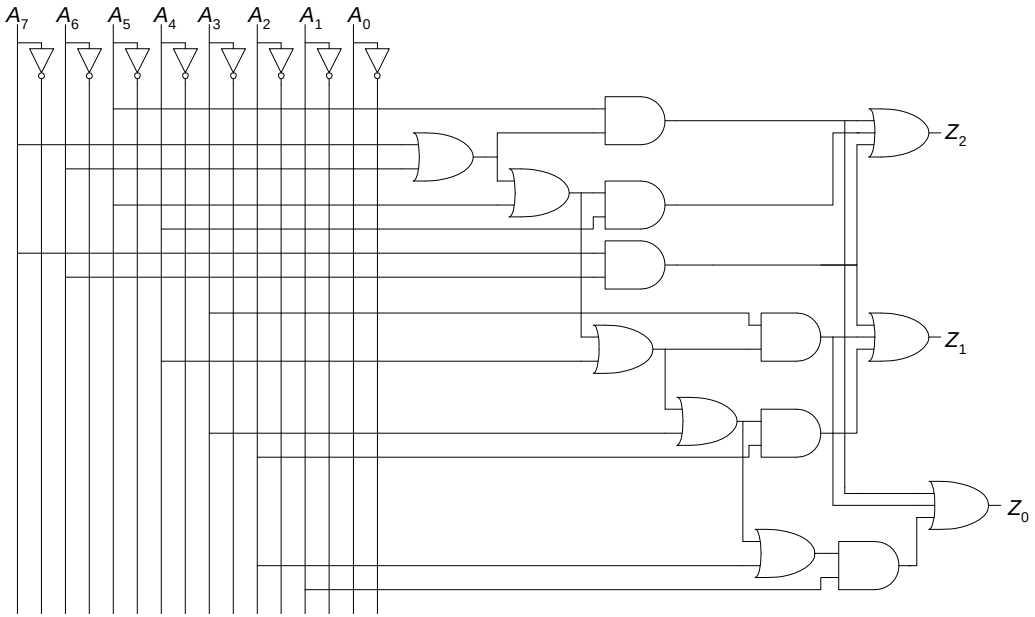
The truth table, equations, and circuit for $Z_{2:0}$ are as follows.

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Z_2	Z_1	Z_0
0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	1	X	0	0	1
0	0	0	0	1	0	1	X	0	0	1
0	0	0	1	0	0	1	X	0	0	1
0	0	1	0	0	0	1	X	0	0	1
0	1	0	0	0	0	1	X	0	0	1
1	0	0	0	0	0	1	X	0	0	1
0	0	0	0	1	1	X	X	0	1	0
0	0	0	1	0	1	X	X	0	1	0
0	0	1	0	0	1	X	X	0	1	0
0	1	0	0	0	1	X	X	0	1	0
1	0	0	0	0	1	X	X	0	1	0
0	0	0	1	1	X	X	X	0	1	1
0	0	1	0	1	X	X	X	0	1	1
0	1	0	0	1	X	X	X	0	1	1
1	0	0	0	1	X	X	X	0	1	1
0	0	1	1	X	X	X	X	1	0	0
0	1	0	1	X	X	X	X	1	0	0
1	0	0	1	X	X	X	X	1	0	0
0	1	1	X	X	X	X	X	1	0	1
1	0	1	X	X	X	X	X	1	0	1
1	1	X	X	X	X	X	X	1	1	0

$$Z_2 = A_4(A_5 + A_6 + A_7) + A_5(A_6 + A_7) + A_6A_7$$

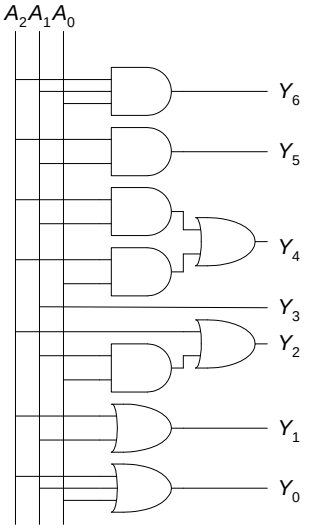
$$Z_1 = A_2(A_3 + A_4 + A_5 + A_6 + A_7) + A_3(A_4 + A_5 + A_6 + A_7) + A_6A_7$$

$$Z_0 = A_1(A_2 + A_3 + A_4 + A_5 + A_6 + A_7) + A_3(A_4 + A_5 + A_6 + A_7) + A_5(A_6 + A_7)$$



Exercise 2.38

$$Y_6 = A_2A_1A_0$$
$$Y_5 = A_2A_1$$
$$Y_4 = A_2A_1 + A_2A_0$$
$$Y_3 = A_2$$
$$Y_2 = A_2 + A_1A_0$$
$$Y_1 = A_2 + A_1$$
$$Y_0 = A_2 + A_1 + A_0$$



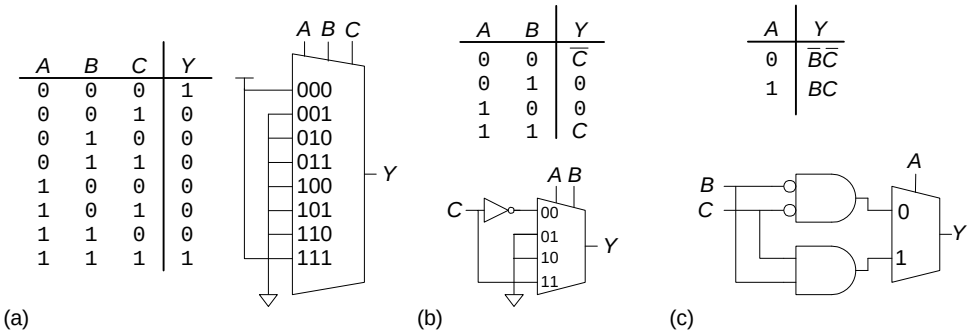
Exercise 2.39

$$Y = A + \overline{C \oplus D} = A + CD + \bar{C}\bar{D}$$

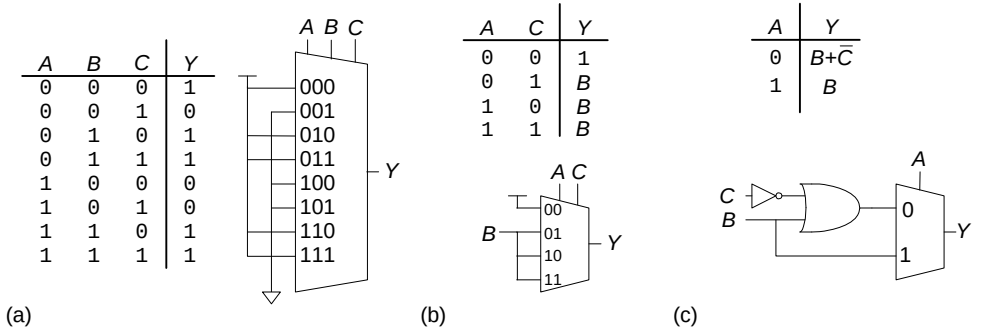
Exercise 2.40

$$Y = \bar{C}\bar{D}(A \oplus B) + \bar{A}\bar{B} = \bar{A}\bar{C}\bar{D} + \bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}$$

Exercise 2.41



Exercise 2.42



Exercise 2.43

$t_{pd} = 3t_{pd_NAND2} =$ **60 ps**
$t_{cd} = t_{cd_NAND2} =$ **15 ps**

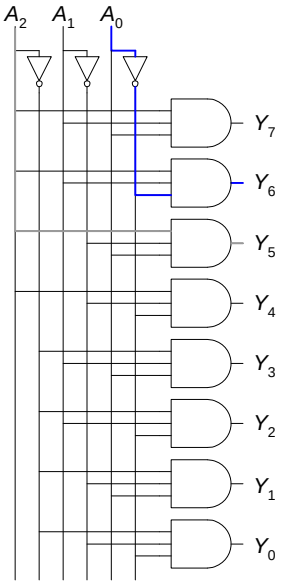
Exercise 2.44

$t_{pd} = t_{pd_AND2} + 2t_{pd_NOR2} + t_{pd_NAND2}$
$= [30 + 2\ (30) + 20]$ ps
$=$ **110 ps**
$t_{cd} = 2t_{cd_NAND2} + t_{cd_NOR2}$
$= [2\ (15) + 25]$ ps
$=$ **55 ps**

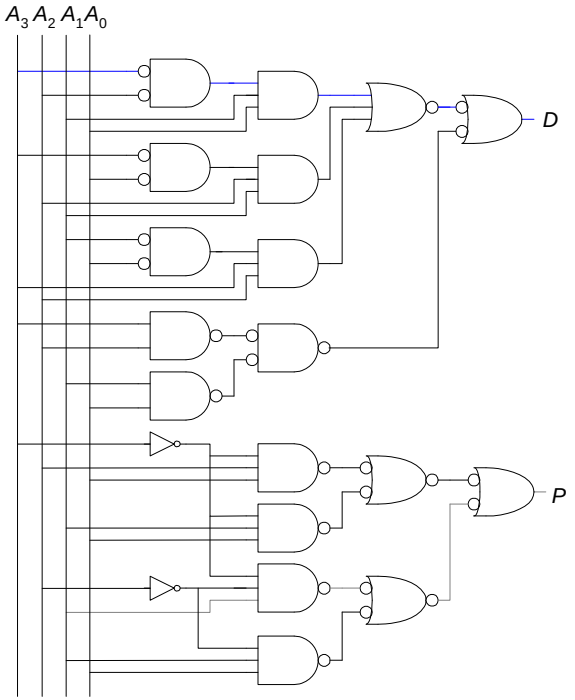
David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc. Exercise Solutions

Exercise 2.45

$t_{pd} = t_{pd_NOT} + t_{pd_AND3}$
$= 15 \text{ ps} + 40 \text{ ps}$
$= \mathbf{55 \text{ ps}}$
$t_{cd} = t_{cd_AND3}$
$= \mathbf{30 \text{ ps}}$



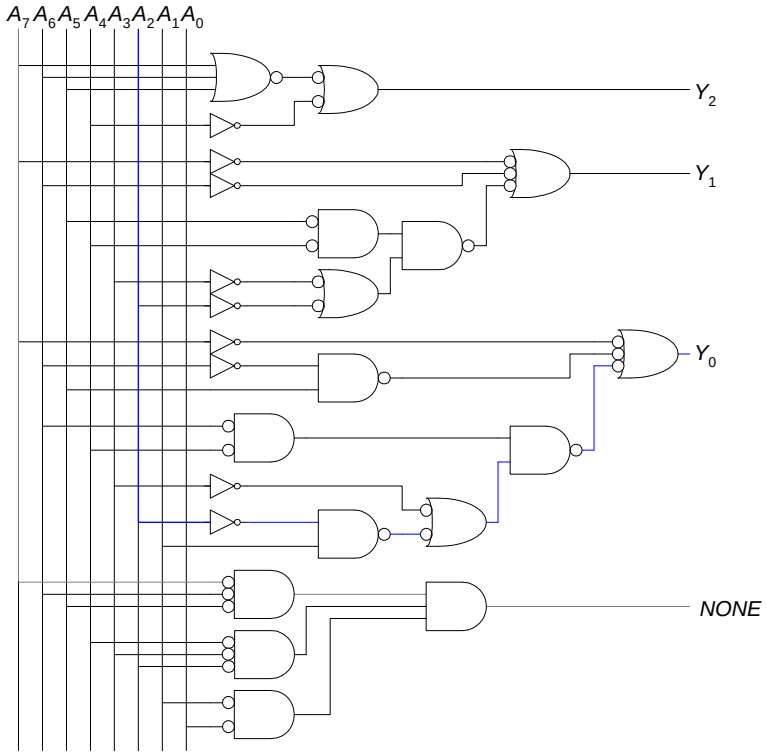
Exercise 2.46



$t_{pd} = t_{pd_NOR2} + t_{pd_AND3} + t_{pd_NOR3} + t_{pd_NAND2}$
$= [30 + 40 + 45 + 20]$ ps
$=$ **135 ps**
$t_{cd} = 2t_{cd_NAND2} + t_{cd_OR2}$
$= [2\ (15) + 30]$ ps
$=$ **60 ps**

Exercise 2.47

Digital Design and Computer Architecture 2nd Edition Harris Solutions Manual
Full Download: http://alibabadownload.com/product/digital-design-and-computer-architecture-2nd-edition-harris-solutions



$t_{pd} = t_{pd_INV} + 3t_{pd_NAND2} + t_{pd_NAND3}$
$= [15 + 3\ (20) + 30]$ ps
$=$ **105 ps**
$t_{cd} = t_{cd_NOT} + t_{cd_NAND2}$
$= [10 + 15]$ ps
$=$ **25 ps**

Exercise 2.48

This sample only, Download all chapters at: alibabadownload.com