CS 4341 — Digital Logic and Computer Design, Homework 1

https://iniyanjoseph.com | iniyanijoseph@gmail.com

1.12 What is the smallest (most negative) 32-bit binary number that can be represented with

Unsigned numbers?

$$0$$

Two's complement numbers?

$$-2^{31}$$

Sign/Magnitude Numbers?

$$-2^{31} - 1$$

1.16 (c,d) Convert the following unsigned binary numbers to hexadecimal. Show your work.

11010111_2

$$1101_0111_2$$
$$= D7_{16}$$

011101010100100_2

$$0011_1010_1010_0100_2$$
$$= 3AA4_{16}$$

1.18c Convert the following hexadecimal numbers to decimal. Show your work.

$ED3A_{16}$

$$= (10 \times 16^0) + (3 \times 16^1) + (13 \times 16^2) + (14 \times 16^3)$$
$$= 10 + 48 + 3328 + 57344$$
$$= 60730_{10}$$

1.20d Convert the following hexadecimal numbers to unsigned binary. Show your work.

$403FB001_{16}$

Because hexadecimal is 2^4, we can simply convert each hexadecimal digit into a 4-bit binary number

$$= 0100_0000_0011_1111_1011_0000_0000_0001$$
$$= 01000000001111111011000000000001_2$$

1.22 (c,d) Convert the following two's complement binary numbers to decimal.

01001110_2

$$= 2^1 + 2^2 + 2^3 + 2^6$$
$$= 78$$

10110101_2

$$= 2^0 + 2^2 + 2^4 + 2^5 + 2^7$$
$$= -75$$

1.24 d Repeat Exercise 1.22, assuming the binary numbers are in sign/magnitude form rather than two's complement representation.

10110101_2

$$= -(2^0 + 2^2 + 2^4 + 2^5 + 2^7)$$
$$= -53$$

1.30 (b,c,e) Convert the following decimal numbers to 8-bit two's complement numbers or indicate that the decimal number would overflow the range.

-59_{10}

$$59 = 29 * 2 + 1; 29 = 14 * 2 + 1; 14 = 7 * 2 + 0; 7 = 3 * 2 + 1; 3 = 1 * 2 + 1; 1 = 2 * 0 + 1$$
$$59 = 00111011$$

After flipping all bits starting at the first one,

$$-59_{10} = 11000101$$

128_{10}

$$10000000_2$$

This is simple, since 128_{10} is a power of 2

-150_{10}

$$50 = 75 * 2 + 0; 75 = 37 * 2 + 1; 37 = 18 * 2 + 1; 18 = 9 * 2 + 0; 9 = 4 * 2 + 1; 4 = 2 * 2 + 0; 2 = 1 * 2 + 0; 1 = 0 * 2 +$$
$$150_{10} = 10010110$$

After flipping all bits starting at the first one,

$$01101010_2$$

1.40 c Convert each of the following octal numbers to binary, hexadecimal, and decimal.

371_8

Because octal is 2^3, we can simply convert each octal digit into a 3-bit binary number

$$011_111_001 = 11111001_2$$

From here, because hexadecimal is 2^4, we can simply convert every 4 binary digits into a decimal number.

$$1111_{1001_2} = \mathrm{F9}_{16}$$
$$\mathrm{F9}_{16} = 9 \times 16^0 + 15 \times 16^1$$
$$249$$

1.42 How many 7-bit two's complement numbers are greater than 0? How many are less than 0? How would your answers differ for sign/magnitude numbers?

2′s complement

$$[-2^6, 2^6 - 1]$$

Sign/magnitude

$$[-2^6 - 1, 2^6 + 1]$$

1.56 (a,c,e) Convert the following decimal numbers to 6-bit two's complement binary numbers and add them. Indicate whether or not the sum overflows a 6-bit result.

$16_{10} + 9_{10}$

$$= 010000_2 + 001001_2$$
$$= 011001_2$$

$-4_{10} + 19_{10}$

$$4_{10} = 100_2$$
$$19_{10} = 9 * 2 + 1; 9 = 4 * 2 + 1; 4 = 2 * 2 + 0; 2 * 1 * 2 + 0; 1 = 0 * 2 + 1$$
$$= 111100_2 + 010011_2$$
$$= 001111_2$$

$-16_{10} + -9_{10}$

$$16_{10} = 010000_2$$
$$9_{10} = 001001_2$$
$$-16_{10} = 110000_2$$
$$-9_{10} = 110111_2$$
$$110111_2 + 110111_2 = 100111_2$$

1.73 A majority gate produces a TRUE output if and only if more than half of its inputs are TRUE. Complete a truth table for the three-input majority gate shown in Figure 1.41.

A	B	C	D
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	T

T	F	F	F
T	F	T	T
T	T	F	T
T	T	T	T

1.82 While walking down a dark alley, Ben Bitdiddle encounters a two-input gate with the transfer function shown in Figure 1.48. The inputs are A and B and the output is Y.

What kind of logic gate did he find?

He found an `AND` gate

What are the approximate high and low logic levels?

The approximate high and low logic levels are 2 and 1.8 respectively.

2.2 (c,d) Write a Boolean equation in sum-of-products canonical form for each of the truth tables

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

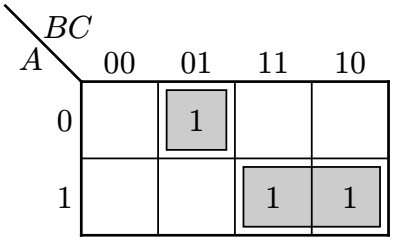
$$f(A,B,C) = \Sigma(m_1, m_6, m_7)$$
$$= \overline{A} \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} + A \cdot B \cdot C$$

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

$$f(A,B,C,D) = \Sigma(m_0, m_2, m_3, m_6, m_7, m_8, m_{10})$$
$$= \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot B \cdot C \cdot \overline{D} + \overline{A} \cdot B \cdot C \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot \overline{D}$$

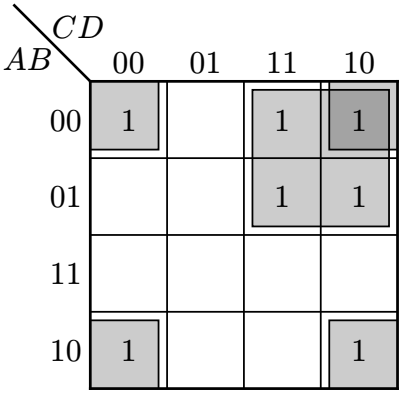
2.6 (c,d) Minimize each of the Boolean equations from Exercise 2.2.

$f(A,B,C) = \Sigma(m_1, m_6, m_7)$



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

$f(A,B,C,D) = \Sigma(m_0, m_1, m_2, m_3, m_8, m_{10}, m_{14})$



$$f(A,B,C,D) = \overline{A} \cdot C + \overline{B} \cdot \overline{D}$$

2.8 (c,d) Sketch a reasonably simple combinational circuit implementing each of the functions from Exercise 2.6. (The problem references the functions from 2.2)

c

d

2.12 (c,d) Repeat Exercise 2.8 using only NOT gates and NAND and NOR gates.

c

d

2.14 (a, b) Simplify the following Boolean equations using Boolean theorems. Check for correctness using a truth table or K-map.

Y = ABC + ABC

$\overline{A}BC + \overline{A}B\overline{C} = \overline{A}B$, since $\overline{A} \cdot B$ remains constant for both, and it doesn't matter if C is true or not.

A	B	C	Y	Not A AND B
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

Y = ABC + AB

$\overline{A \cdot B \cdot C} + A \cdot \overline{B} = \overline{A} \cdot \overline{B} \cdot \overline{C}$. This is easy to see directly from the truth table, since only when all values are 1 is Y = 0. This can be written in product of maxterms form, which is clearly simplified.

A	B	C	Y	NOT A OR NOT B OR NOT C
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

2.24 Write Boolean equations for the circuit in Figure 2.82. You need not minimize the equations.

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

$$Z = B \cdot D + A \cdot \overline{C} \cdot D$$

2.26 Using De Morgan equivalent gates and bubble pushing methods, redraw the circuit in Figure 2.83 so that you can find the Boolean equation by inspection. Write the Boolean equation.

$$\left(\overline{A} + \overline{B}\right)\left(\overline{C} + \overline{D}\right) + \overline{E}$$

2.28 Find a minimal Boolean equation for the function in Figure 2.85. Remember to take advantage of the don't care entries.

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

Let's make a k-map

AB \ CD	00	01	11	10
00	X	X		X
01		X	X	
11	1	1	1	X

10	1		1	X

This means that we get:

$$f(A, B, C, D) = A \cdot \overline{D} + A \cdot C + B \cdot D$$

2.36 A priority encoder has 2N inputs. It produces an N-bit binary output indicating the most significant bit of the input that is TRUE, or 0 if none of the inputs are TRUE. It also produces an output NONE that is TRUE if none of the inputs are TRUE. Design an eight-input priority encoder with inputs A7:0 and outputs Y2.0 and NONE. For example, if the input is 00100000, the output Y should be 101 and NONE should be 0. Give a simplified Boolean equation for each output, and sketch a schematic.

A7	A6	A5	A4	A3	A2	A1	A0	Y2	Y1	Y0	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

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

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

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

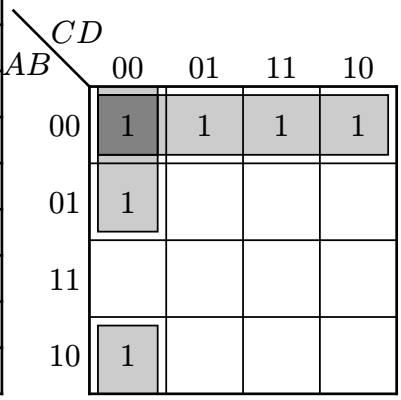
$$\text{NONE} = \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$$

2.40 Write a minimized Boolean equation for the function performed by the circuit in Figure 2.88.

(NOT A AND NOT B) OR (NOT A AND B AND NOT C AND NOT D) OR (A AND NOT B AND NOT C AND NOT D)

$$\overline{A} \cdot \overline{B} + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$$

A	B	C	D	F
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	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
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



$$\overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{C} \cdot \overline{D}$$

2.42 (a, b) Implement the function from Exercise 2.17(a) using

$Y = BC + B\overline{C} + \overline{A}\overline{B}\overline{C}$

An 8:1 multiplexer

A 4:1 multiplexer and no other gates

Simplify the equation so

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

2.44 Determine the propagation delay and contamination delay of the circuit in Figure 2.84. Use the gate delays given in Table 2.8.

The Propagation delay is 110 because of the longest path along the bottom is $30 + 30 + 30 + 20$. The contamination delay is $25 + 15 + 15 = 55$ along the upper path, since it is the shortest weight path to Y.
Used https://www.circuit-diagram.org/editor/ and https://logic.ly/demo/ for diagrams and typst for typsetting