Dhruv Thippasandra
NetID-dthip001

CS61 Homework 3

READING: Chapter 2 (to end)
Read ahead for Week 3: Chapter 3, sections 3.1 through 3.3

HOMEWORK SET (11 questions):
Do the following exercises from Chapter 2 of the text:

Exercises 2.26; 2.30; 2.33 through 2.36; 2.39 - 2.40; 2.43; 2.47; 2.50
Note: 2.39 (c) - just go to 3 binary places of precision, i.e. to the nearest ⅛

3.1

	N-type	P-type
Gate = 1	closed	open
Gate = 0	open	closed

3.2
Replace the missing parts in the following circuit with either a wire or no wire to give the output OUT a logical value of 0 when the input IN is a logical 1.

Replace the circuit with P Type Metal Oxide Semiconductor for the top and the N Type Semiconductor for the bottom.

3.3
A two-input AND and a two-input OR are both examples of two-input
logic functions. How many different two-input logic functions are
possible?

0 0
0 1
1 0

There are 3 different two-input logic functions are possible

2.26

You wish to express −64 as a 2’s complement number.

-64=1100 0000
a. How many bits do you need (the minimum number)?
Need 8 bits
b. With this number of bits, what is the largest positive number you can represent? (Please give answer in both decimal and binary.)
0111 1111
Which is 127 in decimal.

c. With this number of bits, what is the largest unsigned number you can represent? (Please give answer in both decimal and binary.)
1111 1111
Which is 255 in decimal.

2.30

Compute the following. Write your results in binary.
a. 01010111 AND 11010111
01010111

b. 101 AND 110
100

c. 11100000 AND 10110100
10100000

d. 00011111 AND 10110100
00010100

e. (0011 AND 0110) AND 1101
0000

f. 0011 AND (0110 AND 1101)
0000

2.33

Compute the following:
a. 01010111 OR 11010111–11010111
b. 101 OR 110–111
c. 11100000 OR 10110100–11110100
d. 00011111 OR 10110100–10111111
e. (0101 OR 1100) OR 1101–1101

f. 0101 OR (1100 OR 1101)–1111

2.34

Compute the following:
a. NOT(1011) OR NOT(1100)--0111
b. NOT(1000 AND (1100 OR 0101))--0111
c. NOT(NOT(1101))--1101
d. (0110 OR 0000) AND 1111--0110

2.35

In Example 2.11, what are the masks used for?

A mask is a binary number that is used to modify or extract specific bits from another binary number. Masks are represented in binary and are composed of a series of 1s and 0s, where 1s denote bits to be kept or modified and 0s denote bits to be ignored or set to 0.

2.36

Refer to Example 2.11 for the following questions.
a. What mask value and what operation would one use to indicate that
machine 2 is busy?

To indicate that machine 2 is busy, we would use the bit mask 01000100 and perform the logical AND operation with the current bit vector.

b. What mask value and what operation would one use to indicate that
machines 2 and 6 are no longer busy? (Note: This can be done with
only one operation.)

To indicate that machines 2 and 6 are no longer busy, we would use the bit mask 10111101 and perform the logical OR operation with the current bit vector.

c. What mask value and what operation would one use to indicate that
all machines are busy?

To indicate that all machines are busy, we would use the bit mask 11111111 and perform the logical AND operation with the current bit vector.

d. What mask value and what operation would one use to indicate that
all machines are idle?

To indicate that all machines are idle, we would use the bit mask 00000000 and perform the logical OR operation with the current bit vector.

e. Using the operations discussed in this chapter, develop a procedure to
isolate the status bit of machine 2 as the sign bit. For example, if the
BUSYNESS pattern is 01011100, then the output of this procedure is
10000000. If the BUSYNESS pattern is 01110011, then the output is
00000000. In general, if the BUSYNESS pattern is:
b7 b6 b5 b4 b3 b2 b1 b0
the output is:
b2 0 0 0 0 0 0 0 .
Hint: What happens when you ADD a bit pattern to itself?

To isolate the status bit of machine 2 as the sign bit, we can perform the logical XOR operation of the current bit vector with the bit mask 00000100. This will give us a new bit vector where only bit 2 is set to 1 and all other bits are set to 0. Then, we can perform the logical OR operation of this new bit vector with the bit mask 10000000 to shift the bit to the leftmost position and obtain the final output.

2.39

Write IEEE floating point representation of the following
decimal numbers.
a. 3.75→0 10000000 11110000000000000000000
b. −55 23/64→1 10000101 10111001011000000000000
c. 3.1415927→0 10000000 1001001000011111101101010001110001110111111110110101
d. 64,000→0 10001110 00111000100000000000000

2.40

Write the decimal equivalents for these IEEE floating point
numbers.
a. 0 10000000 00000000000000000000000→0
b. 1 10000011 00010000000000000000000→-55.75
c. 0 11111111 00000000000000000000000→10.141592
d. 1 10000000 10010000000000000000000→-64.0

2.43

Translate the following ASCII codes into strings of characters by
interpreting each group of eight bits as an ASCII character.
a. x48656c6c6f21→Hello!
b. x68454c4c4f21→HELLO!
c. x436f6d70757465727321→Computers!

d. x4c432d32→LC-2

2.47

Convert the following hexadecimal representations of 2’s complement binary numbers to decimal numbers.
a. xF0→11110000→-112
b. x7FF→011111111111→2047
c. x16→00010110→22
d. x8000→1000000000000000→0

2.50

Perform the following logical operations. Express your answers in hexadecimal notation.
a. x5478 AND xFDEA→x5078

b. xABCD OR x1234→xBFFF

c. NOT((NOT(xDEFA)) AND (NOT(xFFFF)))--> x2105

d. x00FF XOR x325C→x32A3