CSU200 Discrete Structures — Profs. Aslam & Fell
Fall 2006 — September 13, 2006

Written Homework 01

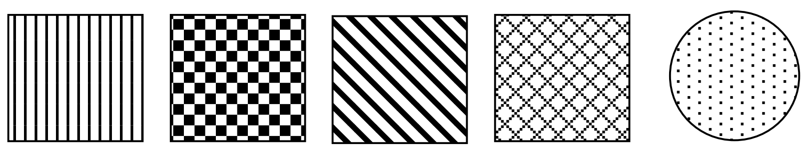
Assigned: Wed 13 Sep 2006
Due: Wed 20 Sep 2006

Instructions:

- The assignment is due at the *beginning* of class on the due date specified, i.e., 1:35pm for Prof. Aslam's section and 4:35pm for Prof. Fell's section. Late assignments will be penalized 50%, as stated in the course information sheet. Late assignments *will not be accepted* after the solutions have been distributed.
- Problem 1 requires graph paper. Printable graph paper may be found at the following URL:

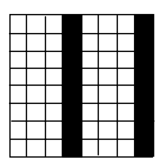
`http://www.pdfpad.com/graphpaper/pdf/c-i-14.pdf`

Problem 1 [24 pts, (16,8)]: Patterns.

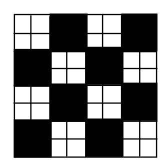


Patterns, like the ones above, are available in most drawing programs for filling regions. A pattern is often defined by an 8×8 array of bits. In each of the following two examples, the 8×8 array of bits on the left corresponds to the pattern on the right.

0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1

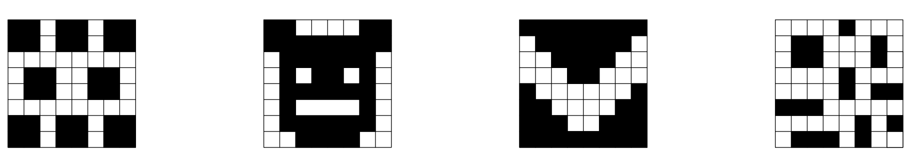


0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	0
0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	0



The 0s represent white, and the 1s represent black. Each row is an 8-bit binary number. As we know, a 4-bit binary number can be expressed as a single hex-digit, so an 8-bit binary number can be expressed with two hex-digits. Designers specify a pattern by giving eight 2-hex-digit numbers, one 2-hex-digit number per row. The two patterns given above are encoded as "11, 11, 11, 11, 11, 11, 11, 11" and "33, 33, CC, CC, 33, 33, CC, CC."

i. For each of the following patterns, give the eight 2-hex-digit encoding.



ii. Use graph paper to show the pattern described by each of the following sequences of eight 2-hex-digit numbers. (See the instructions above for a link to printable graph paper.)

28, 6A, 31, 77, 77, 13, A6, 82   CE, B4, EC, 4B, CE, B4, EC, 4B

Problem 2 [24 pts, (8,8,8)]**:** Negative numbers and two's complement.

i. Give the 8-bit two's complement representations of the following integers: 34, 66, -71, -27.

ii. Give the integer (in standard base-10 notation) which is represented by each of the following 8-bit two's complement numbers: 01100110, 10011001, 01010101, 11011101.

iii. Compute the following sums and differences using 8-bit two's complement representations, as shown in class and described in the text: $66 - 27$, $-71 - 27$. Verify that your answers are correct by converting the results back to standard base-10 notation.

Note: Use the two's complement representations from part i above.

Problem 3 [32, (8,8,8,8)]**:** Multiplication.

Perform the following multiplications in binary. For each problem part, you must (1) convert each decimal number to binary, (2) perform the multiplication in binary, and (3) convert the binary result back to decimal. You must show your work.

Note: For consistency, place the binary representation of the left multiplicand in the top row of your multiplication and place the binary representation of the right multiplicand on the bottom row of your multiplication. Thus, "4×7" would be

$$\begin{array}{cccc} & 1 & 0 & 0 \\ \times & 1 & 1 & 1 \\ \hline \end{array}$$

while "7×4" would be

$$\begin{array}{cccc} & 1 & 1 & 1 \\ \times & 1 & 0 & 0 \\ \hline \end{array}$$

by this convention.

i. 27×6

ii. 23×11

iii. 11×23

iv. 46×7

Problem 4 [20, (4,4,12)]**:** Logical completeness.

Every truth table, Boolean formula, and circuit can be implemented using just AND, OR, and NOT gates; hence, the collection {AND, OR, NOT} is *logically complete.* In this problem, you will show that the NAND gate, by itself, is logically complete. To do so, you will show how to construct the AND, OR, and NOT gates from NAND gates.

i. Fill in the following truth table:

X	X NAND X
0	?
1	?

What logical operation does X NAND X correspond to?

ii. Fill in the following truth table:

X	Y	$\neg X$ NAND $\neg Y$
0	0	?
0	1	?
1	0	?
1	1	?

What logical operation does $\neg X$ NAND $\neg Y$ correspond to?

iii. Using only NAND gates, draw circuit diagrams corresponding to the AND, OR, and NOT gates. *Hint:* The constructions for two of these circuits are essentially given in parts i and ii above, and the construction of the third should be relatively straightforward given what you've learned above.