Homework # 1
Due Thur., Sept 18, 2009, at 9:45 AM in class. No late homeworks will be accepted except for verifiable illness or similar situations.

1. Conversion Between Different Number Representations

(a) Convert the following binary numbers to decimal:
   i. 111011
   ii. 10111.0101
   iii. 101.10101

(b) Convert the following decimal numbers to binary:
   i. 78996
   ii. 0.007778
   iii. 968.277

(c) Perform the following number conversions:
   i. Binary 101011.11101 to decimal, octal and hexadecimal.
   ii. Octal 53.4 to decimal, binary and hexadecimal.
   iii. Hexadecimal F5.BB2 to decimal, binary and octal.

(d) Perform the following number conversions:
   i. Binary 1010111010101010101010001011111101000111111 to hexadecimal
   ii. Hexadecimal AF87BC69758EF67978798 to binary.

(e) Give the negative of the following numbers in two’s complement notation.
   i. 00000000
   ii. 01111011
   iii. 11010101
   iv. 11011101
(f) Suppose that the following bits represent positive and negative numbers in two’s complement notation. Perform the operation indicated. Give the result in decimal.

i. \(10001000_2 + 01111001_2\).

ii. \(00111110_2 - 01001110_2\).

iii. \(00001001_2 \times 00001010_2\).

iv. \(10101010_2 \times 11111111_2\).

2. Truth Tables from Functions

Write down the truth tables for the following functions.

(a) \(f(a, b, c, d) = ab\overline{c} + a\overline{c} + a\overline{d} + b\overline{d} + c\overline{d}\)

(b) \(g(a, b, c, d) = ac(\overline{b} + \overline{c}) + d(\overline{b}\overline{c} + \overline{a})\)

3. Seven-Segment Display

Consider the example shown in Figure 1: a 7-segment display decoder. The inputs are four bits, \(x_0, x_1, x_2, x_3\), specifying a number in binary. The outputs are 7 bits, \(a, b, c, d, e, f, g\), specifying which segments to light up in a display – such as that of a digital alarm clock – to form the image of the corresponding decimal number. Our goal is to design a circuit that implements the functions \(a, b, c, d, e, f, g\) (with 0 corresponding to “off” and 1 corresponding to “on”).

Since the four bits \(x_0, x_1, x_2, x_3\) specify numbers from 0 to 15, but the display is only for 0 through 9, we’ll adopt the convention that for inputs corresponding to 10, 11, 12, 13, 14 and 15 all the segments should light up (i.e., \(a\) through \(g\) evaluate to 1 for these inputs).

\[\begin{array}{cccccc}
& & c & & \\
& a & & d & f \\
b & & e & & g \\
& & & & \\
\end{array}\]

Figure 1: 7-Segment Display.
(a) **Truth Table Design**

Fill in the following truth table.

<table>
<thead>
<tr>
<th>$x_3$</th>
<th>$x_2$</th>
<th>$x_1$</th>
<th>$x_0$</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$e$</th>
<th>$f$</th>
<th>$g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) **Two-Level Expressions**

i. Produce a AND-OR (i.e., sum-of-products) expression for $a, b, c, d, e, f$ and $g$.

ii. Produce a OR-AND (i.e., product-of-sums) expression for $a, b, c, d, e, f$ and $g$.

iii. Produce a AND-XOR expression for $a, b, c, d, e, f$ and $g$ (with no negations).