EECS 16B Designing Information Devices and Systems II
Spring 2016 Anant Sahai and Michel Maharbiz Discussion 4A

1. **Binary Encoding Review**

   We have used decimal numbers all our lives. Here we want to introduce a different way to express numbers: binary encoding. The term *bit* means "binary digit", meaning 0 or 1 in binary encoding. In a binary number, a bit's value depends on its position, starting from the right. Like tens, hundreds, and thousands in a decimal number, a bit's value grows by a power of two as its position goes from right to left. Here we use $(N)_2$ to indicate N is expressed by binary encoding, and $(N)_{10}$ means N is expressed in decimal. For example, $(245)_{10} = 2\times 10^2 + 4\times 10^1 + 5\times 10^0$, while $(11010)_2 = 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0$.

   (a) Express $(10)_{10}$ with binary encoding.

   (b) Given k binary digits, what is the biggest number you can express?

2. **Boolean Formulae And Natural Numbers** In Boolean algebra, the values of all variables are the truth values, *true* and *false*, usually denoted 1 and 0 respectively. A Boolean formula is like an arithmetic combination of a set of Boolean variables, $\{B_0, \cdots B_n\}$, with Boolean operators, **AND**, **OR**, etc. Here we use $\neg$ for **NOT**, $\wedge$ for **AND**, $\vee$ for **OR**, and $\oplus$ for **XOR**. Notice that $\neg$ is a unary operator which only acts on one Boolean variable, while others are binary operators.

   A Boolean formula can evaluate to a truth value, once the values of all variables in this formula are defined. For example, a Boolean formula, $f = B_0 \wedge B_1$, shows a Boolean operator, **AND**, acting on two Boolean variables, B_0 and B_1. If $B_0 = B_1 = 1$, f evaluates to 1, while $f = 0$ under other values of B_0 and B_1. Another Boolean formula $g = \neg B_0$ evaluates to 1 when $B_0 = 0$, $g = 0$ when $B_1 = 1$.

   In this question, we want to show that each natural number can be expressed by a Boolean formula with a set of Boolean variables representing its binary digits. For a binary number $(10)_2$, suppose we have two Boolean variables, B_0 and B_1 to express the two bits of this number. We want to check if they express this number correctly, so we need to find a Boolean formula h of B_0 and B_1, such that when the two variables express $(10)_2$ correctly, h evaluates to 1. In this case, h should be $B_1 \wedge \neg B_0$, if we use B_0 to express the rightmost bit of this number. Hence h will evaluate to 1 only when $B_0 = 0$ and $B_1 = 1$.

   (a) Express an integer, $(1)_{10}$, with a Boolean formula f, which includes only one Boolean variable B_0.

   (b) Express an integer $(0)_{10}$ with a Boolean formula g, which includes only one Boolean variable B_0.

   (c) Express an integer $(17)_{10}$, with a Boolean formula h, with 5 Boolean variables, B_0, B_1, B_2, B_3 and B_4.

   (d) Given a natural number $(N)_{10}$, how could we express it with a Boolean formula? How many Boolean variables do we need?

3. **Binary Addition with Boolean Operators** Here we want to show how to perform addition between two binary numbers.

   (a) Compute $(1)_2 + (1)_2$ in binary encoding.

   (b) Compute $(1111)_2 + (1001)_2$.

(c) Now let's think about how to express binary addition with Boolean operators. Suppose we want to perform binary addition between two Boolean variables, A_0 and B_0, we should have two Boolean variables to express the results of carry (C_0) and sum (S_0). Express the two outputs with Boolean formulae in terms of A_0 and B_0.

(d) Now we want to perform binary addition between two 2-bit binary numbers, $(A_1A_0)_2$ and $(B_1B_0)_2$. . Could we reuse the results from (c)? *(Hint: could we use C_0 to express some formulae?)*

(e) Use the results above to verify $(10)_2 + (11)_2 = (101)_2$.

Contributors:

- Yu-Yun Dai.