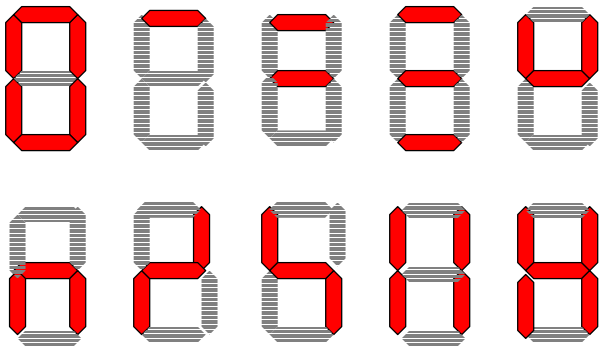
University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 150 R. H. Katz
Fall 2000

Problem Set # 4 (Assigned 21 September, Due 29 September)

1. Your task is to design a combinational logic subsystem as part of a larger system that makes change from quarters. There is a large reservoir of dimes and another one of nickels. There are also two binary counters that keep track of the number of dimes and nickels in each (these are outside your subsystem, and you won't need to design them—but will need them as inputs). Your subsystem needs to work as follows. It takes the low order bits of the dimes counter and the nickels counter, and generates the number of dimes to give as change and the number of nickels. In general, dimes should be given before nickels, e.g., if there are at least two dimes left, these should be given with a single nickel, rather than five nickels. There is definitely a possibility that no change can be given before the reservoirs are (almost) exhausted of coins, and a "no change" sign should be illuminated.
   (a) Identify your inputs and outputs.
   (b) Specify the encoding of your outputs: what do they mean?
   (c) Develop the minimized gate-level implementation using the K-map method.
2. Your task is to design a combinational logic subsystem to decode a BCD digit in the range of 0 through 9 in order to drive a seven segment display for the Klingon number system (note: Klingons fortunately have ten figures). The Klingon numerals are as follows:



   Design a minimized implementation in PLA form. That is, look for common terms among the seven output functions.
3. Consider the following logic unit. It has three operation inputs (A, B, C), two data inputs (D1, D0), and a single output (Z). The logic unit is defined as follows: when ABC=000, Z= 0; ABC=001, Z=D0; ABC=010, Z=D1 AND D0; ABC=011, Z=D1 NAND D0; ABC=100, Z=D1 NOR D0; ABC=101, Z=D1 OR D0; ABC=110, Z=D1; ABC=111, Z=1.
   (a) Implement to obtain a minimized implementation using the K-map method (NOTE: this uses 5 variables!).
   (b) Implement using a 16:1 Mux (use A,B,C,D1 as your MUX control inputs).
   (c) Implement using an 8:1 Mux plus whatever additional gate logic you need (use A,B,C as your MUX control inputs).
   (d) Which implementation is "best"? Why?