COS126: General Computer Science **Spring 1996**

Exercise # 7

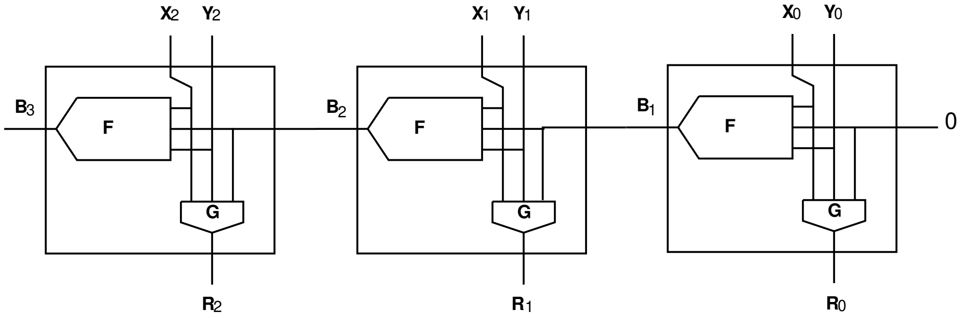
Handed out:April. 23, 1996

These exercises are meant to guide class discussion. *Do not* turn them in.

1. Draw a Boolean circuit using AND, OR, and NOT gates that implements the function $XY + XZ$.
2. Draw a Boolean circuit using AND, OR, and NOT gates that implements the function $X(Y + Z)$.
3. Draw a Boolean circuit using AND, OR, and NOT gates that implements the function of three variables that is 0 if precisely two of the variables have the same value, 1 otherwise.
4. Draw a Boolean circuit using AND, OR, and NOT gates that implements the function of four variables that is 0 if precisely two of the variables have the same value, 1 otherwise.
5. A NOR gate is a logic element with two inputs that produces 1 if the inputs are both 0, 0 otherwise. Logic circuits can be built using NOR gates (no other kind of gate). Draw a logic circuit that uses NOR gates to implement an AND circuit (output 1 if both inputs are 1, 0 otherwise).
6. What is the result of computing the AND of all the output lines of a decoder? Answer the same question for OR.
7. Subtraction $(x - y)$ of two positive n-bit integers, x and y, can be implemented by changing the majority and parity circuits of the lecture notes (Lecture 19) to two new circuits, F and G. Let $x = x_{n-1}...x_1x_0$, where x_0 is the lowest order bit, and similarly y=$y_{n-1}...y_1y_0$. Circuit G computes bit i of the result, R_i, from x_i, y_i, and B_i, where B_i is the "borrow" needed from x_i for the computation of R_{i-1}. Circuit F computes B_i from x_{i-1}, y_{i-1} and B_{i-1}. The "borrow" is just what you would expect from elementary school days, e.g. if $x_0 = 0$ and $y_0 = 1$, $B_1 = 1$ because we need to borrow '1' from the next higher order bit to make $x_0 = 10$ so we can subtract $y_0 = 1$ (getting $R_0 = 1$).

   The Boolean equations for B_i are:

   $B_0 = 0$
   $B_{i+1} = \overline{x_i}B_i + x_iy_iB_i + \overline{x_i}y_i$



2 What is the Boolean equation for R_i?

b Give an implementation of circuit F computing B_{i+1} from x_i, y_i, and B_i in terms of AND gates, OR gates, and inverters.

c What is the meaning of $B_n = 1$?