1. Reading: K. Rosen *Discrete Mathematics and Its Applications*, 9.3
2. The main message of this lecture:

A computer is made up of circuits which can be described using Boolean algebras.

The basic elements of circuits are called **gates**. The Boolean gates are the **inverter**, the *OR* gates with multiple inputs, and the *AND* gates with multiple inputs (see slides). In a circuit some gates may share inputs.

Examples 38.1. Circuits for given Boolean expressions (see slides): **majority voting**, **light controls**, **adders**.

Examples 38.2. Simplifying Boolean expressions minimizes Boolean circuits. This is where Boolean algebra comes into play. For example. using the properties of Boolean algebra, we can reduce the sum-of-products form

$$x_1x_2x_3 + x_1\overline{x_2}x_3 + x_1\overline{x_2}\ \overline{x_3} + \overline{x_1}\ \overline{x_2}x_3 + \overline{x_1}\ \overline{x_2}\ \overline{x_3}$$

to $x_1x_3 + \overline{x_2}$ as follows:

$$\begin{aligned}
&x_1x_2x_3 + x_1\overline{x_2}x_3 + x_1\overline{x_2}\ \overline{x_3} + \overline{x_1}\ \overline{x_2}x_3 + \overline{x_1}\ \overline{x_2}\ \overline{x_3} = \\
&x_1x_2x_3 + x_1\overline{x_2}x_3 + x_1\overline{x_2}x_3 + x_1\overline{x_2}\ \overline{x_3} + \overline{x_1}\ \overline{x_2}x_3 + \overline{x_1}\ \overline{x_2}\ \overline{x_3} = \\
&x_1x_3x_2 + x_1x_3\overline{x_2} + x_1\overline{x_2}x_3 + x_1\overline{x_2}\ \overline{x_3} + \overline{x_1}\ \overline{x_2}x_3 + \overline{x_1}\ \overline{x_2}\ \overline{x_3} = \\
&x_1x_3(x_2 + \overline{x_2}) + x_1\overline{x_2}(x_3 + \overline{x_3}) + \overline{x_1}\ \overline{x_2}(x_3 + \overline{x_3}) = \\
&x_1x_3 \cdot 1 + x_1\overline{x_2} \cdot 1 + \overline{x_1}\ \overline{x_2} \cdot 1 = \\
&x_1x_3 + x_1\overline{x_2} + \overline{x_1}\ \overline{x_2} = \\
&x_1x_3 + \overline{x_2}x_1 + \overline{x_2}\ \overline{x_1} = \\
&x_1x_3 + \overline{x_2}(x_1 + \overline{x_1}) = \\
&x_1x_3 + \overline{x_2} \cdot 1 = \\
&x_1x_3 + \overline{x_2}
\end{aligned}$$

Unfortunately, applying Boolean algebra properties to simplify an expression remains a fairly creative process, though there are some systematic approaches to this minimization problem. For now, we should say a bit more about why we would want to minimize. In good old times when logic networks were built from separate gates and inverters, the cost of those elements was a considerable factor in the design. It was desirable to have as few elements as possible. Since the early 1960's, however, most networks are built using integrated circuit technology. An integrated circuit is itself a logic network representing a certain truth function or functions. Because the integrated circuits are extremely small and relatively inexpensive, it might seem pointless to bother minimizing a network. However, minimization is still important because of the *reliability* of the final network is a function of the number of connections between the integrated circuit packages.

Moreover, the designers of integrated circuits are highly interested in the minimization problem. The wiring channels required to connect components on silicon chips may be so numerous that the wiring takes up more of the chip's floor space than the components themselves. Minimizing the number of components and the amount of wiring required to realize a desired truth function makes the chip less crowded and easier to design physically. Minimization also makes it possible to embed more functions in a single chip.

Examples 38.3. Programmable Logic Arrays (PLA's). A PLA is a chip that is already implanted with as standard array of *AND* gates and an array of *OR* gates, together with a rectangular grid of wiring channels, and some inverters (see the slide). One Boolean expressions in sum-of-products form have been determined for the truth functions, the required components in the PLA are activated. Although this chip is not very efficient and is practical only for smaller scale circuit logic, the PLA can be mass-produced, and only a small amount of time (i.e. money) is then required to program it for the desired functions. The picture below shows the same PLA programmed to produce the truth functions

$f_1(x_1, x_2, x_3) = x_1x_2x_3 + x_1\overline{x_2}x_3 + x_1\overline{x_2}\ \overline{x_3} + \overline{x_1}\ \overline{x_2}x_3$ and

$f_2(x_1, x_2, x_3) = x_1x_2x_3 + x_1\overline{x_2}x_3 + x_1\overline{x_2}\ \overline{x_3} + \overline{x_1}\ \overline{x_2}x_3 + \overline{x_1}\ \overline{x_2}\ \overline{x_3}$

The dots represent activation points.

Homework assignments. (The third installment due Friday 05/04)
38A:Rosen9.3-4; 38B:Rosen9.3-6bc; 38C:Rosen9.3-8.