Lecture 12 Boolean Functions and Circuit Design

In elementary algebra (or calculus) we had started from the simplest linear function $f(x) \equiv x$ (x is the argument of the function, our real variable) and using the operations of multiplication and summation have constructed polynomial functions

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$$

more complicated functions $(f(x) = \ln x, f(x) = \sin^2 x$ etc.) can be approximated by polynomials. Calculus has the special methods to construct such approximation (Taylor series and so forth).

Of course, in many cases we use the functions in particular polynomial functions of many real variables:

$$f(x_1, x_2) = x_1^2 + x_2^2,\ f(x_1, x_2, x_3) = x_1 \cdot x_2 \cdot x_3 ...$$

In mathematical logic (and more general, in computer sciences) especially important are the *Boolean functions* of *Boolean variables.* A *Boolean variable* x has only two values: 0 and 1 (corresponding to "false" or "true" for some logical sentence or proposition). If we have n Boolean variables $x_1, x_2, ..., x_n$ then the Boolean function $f(x_1, x_2, ...x_n)$ (with only two possible values, 0 or 1) represents the compound sentence constructed in terms of elementary sentences $x_i,\ i = 1, 2, ..., n$. The simplest method to describe such a function is to give the corresponding table (like fig 1)

x_1	x_2...	x_n	$f(x_1, x_2, ...x_n)$
0	0...	0	1
0	0	1	0
0	1	0	1
1	1	1	0

Fig 1

Input of the table contains 2^n rows (product rule, each variable has two different values), output contains an arbitrary sequence of 0's and 1's (possible values of the function f for different combinations of variables values).

Total number of Boolean functions of n variables $x_1, ..., x_n$ is equal 2^{2^n} (because the right column of the table contains 2^n positions and two opportunities for each position).

There are 4 Boolean functions of one variable x:

a) $f(x) = x$

x	f
0	0
1	1

b) $f(x) = \widetilde{x}$

x	f
0	1
1	0

c) $f(x) \equiv 0$

x	f
0	0
1	0

d) $f(x) \equiv 1$

x	f
0	1
1	1

Most interesting is the second function

$$f(x) = \widetilde{x}$$

known as *complement* of x or *negation* of x. Sign "$\sim$" is of course the analogue of the complement operator "$\overline{\circ}$" in the set theory. In the literature one can find the different notations: $x', \sim x$ etc.

Operation of the negation can be realized with computers by the special electronic devices called "*in vectors*" (fig 2)

x $\widetilde{x}$ Fig. 2.

The input of the inverter is an electrical current (magnetic field) with one of two possible direction (voltages) indicated by 0 or 1. The inverter reverses the direction of the current (field) and generates the output $\widetilde{x}$.

The number of the Boolean functions of two variables is already very large: $N_2 = 2^{2^2} = 16$.

Some of these functions are not interesting, for instance

$$\begin{aligned} f_1(x_1, x_2) &\equiv 0, \;\; f_2(x_1, x_2) \equiv 1, \;\; f_3(x_1, x_2) = x_1, \\ f_4(x_1, x_2) &= x_2, \;\; f_5(x_1, x_2) = \widetilde{x_1}, \;\; f_6(x_1, x_2) = \widetilde{x_2} \end{aligned}$$

All these functions are in fact the functions of 1 variable.

Two functions of two variables play the fundamental role. They describe the logical operations, disjunction and conjunction, which we discussed already in the context of the relations theory.

By the definition, the function $f(x_1, x_2) = x_1 \vee x_2$ (disjunction of x_1 and x_2) is given by the table

x_1	x_2	$x_1 \vee x_2 = f(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	1

The meaning of this function is simple:

$$x_1 \vee x_2 = 1 \Leftrightarrow \text{ or } x_1 = 1 \text{ or } x_2 = 1 \text{ or both } x_1 = x_2 = 1.$$

There is a special device ("*or gate*") which transforms the pair of Boolean variables x_1, x_2 in the disjunction x_1, x_2. The corresponding scheme is presented by fig 3:

The dual operation is a *conjunction* of x_1 and x_2.

x_1	x_2	$x_1 \wedge x_2 = f$
0	0	0
0	1	0
1	0	0
1	1	1

It is clear that $x_1 \wedge x_2 = 1 \Leftrightarrow x_1 = 1$ and $x_2 = 1$. This operation can be realized by "*and gate*":

Using operations $\vee$ and $\wedge$ one can construct 8 different functions of two variables:

$f_7(x_1, x_2) = x_1 \vee x_2$	$f_8(x_1, x_2) = x_1 \wedge x_2$
$f_9(x_1, x_2) = \widetilde{x}_1 \vee x_2$	$f_{10}(x_1, x_2) = \widetilde{x}_1 \wedge x_2$
$f_{11}(x_1, x_2) = x_1 \vee \widetilde{x}_2$	$f_{12}(x_1, x_2) = x_1 \wedge \widetilde{x}_2$
$f_{13}(x_1, x_2) = \widetilde{x}_1 \vee \widetilde{x}_2$	$f_{14}(x_1, x_2) = \widetilde{x}_1 \wedge \widetilde{x}_2$

All the functions in the left column have three times the value 1 and one time the value 0, in the right column three times the value 0 and one time the value 1.

Examples.

x_1	x_2	$f_{11}(x_1, x_2) = x_1 \vee \widetilde{x}_2$
0	0	1
0	1	0
1	0	1
1	1	0

and $f_{16}(x_1, x_2) = (\widetilde{x}_1 \wedge \widetilde{x}_2) \vee (x_1 \wedge x_2)$.

We have now the complete list of all 16 Boolean functions of two (Boolean) variables. Let's remark that the representation of these functions is not unique. For instance:

$$\begin{aligned} f(x_1, x_2) &= (x_1 \wedge x_2) \vee (\widetilde{x}_1 \wedge x_2) = \\ &= (x_1 \vee \widetilde{x}_2) \wedge x_2 = \text{(distributive property of conjunction!)} \\ &= 1 \wedge x_2 = x_2 \end{aligned}$$

It is clear that for $n = 3, n = 4$ the number of possible Boolean functions is huge: $N_3 = 2^{2^3} = 2^8 = 256,\ N_4 = 2^{16} = (256)^2 = 65536$. In this case the complete analysis of all possible Boolean functions is complicated. Nevertheless, some general statements can be done.

Definition Boolean polynomial of n variables $x_k,\ k = 1, 2, ...n$ ($x_k = 0$ or 1) is an arbitrary expression $P_n(x_1, ..., x_n)$ containing variables x_k and operations $\vee, n, \sim$.

More precisely. All functions $x_1, x_2, ...x_n$ are the Boolean polynomials, symbols 0 and 1 are the Boolean polynomials (constants). If $B(x_1, ...x_n)$ is a Boolean polynomial then $\widetilde{f}(x_1, ...x_n)$ is also a Boolean polynomial, if $f(x_1, ...x_n),\ g(x_1, ...x_n)$ are two Boolean polynomials, then so are $(f \vee g)(x_1, ...x_n), (f \wedge g)(x_1, ...x_n)$. The simplest Boolean polynomials are the *monomials* (or *Minterms*). These polynomials are the conjunctions of the variables x_k or $\hat{x}_k, k = 1, 2, ...n$.

Say

$$f(x_1, x_2, ...x_n) = x_1 \wedge \widetilde{x}_2 \wedge x_3 \wedge \widetilde{x}_4 ... \wedge x_n$$

There are 2^n monomials with n variables (You have to remember, that $x \wedge x = x$ and $x \wedge \widetilde{x} = 0$).

Each monomial has a very important property: it is equal to 1 only for one specific combination of variables. (In the expression above $f = 0 \Leftrightarrow x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0, ...x_n = 1$).

Theorem. Each Boolean function of n variables can be expressed in the canonical form (disjunctive normal form), i.e. as a disjunction of monomials.

The proof is simple and constructive. Function $f(x_1, ...x_n)$ is given by the table (like at the Fig. 1). Identify each row of the table for which the output is equal to 1. For each such row construct the monomial which has value 1 for the exact combination of input values (take x_k if we have 1 at the k$^{\text{th}}$ position, or $\widetilde{x}_k$ if we have 0 and combine these variables by $\wedge$). At the final step construct the disjunction of monomials.

Example. A Boolean function of three variables is given by the following table

x_1	x_2	x_3	f
0	0	0	0
0	1	0	1
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0
1	0	1	0
1	1	1	1

Construct the corresponding disjunctive normal form.

Solution. Output f is equal to 1 for the second, fourth and eighth rows. Corresponding monomials are equal to

$(\widetilde{x}_1 \wedge x_2 \wedge \widetilde{x}_3)$ (for the second row)
$(\widetilde{x}_1 \wedge x_2 \wedge x_3)$ (for the fourth row)
$(x_1 \wedge x_2 \wedge x_3)$ (for the last row)

As a result

$$f(x_1, x_2, x_3) = (\widetilde{x}_1 \wedge x_2 \wedge \widetilde{x}_3) \vee (\widetilde{x}_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3)$$

This expression, however, is not the simplest one. One can note that monomials

$$(\widetilde{x}_1 \wedge x_2 \wedge \widetilde{x}_3),\ (\widetilde{x}_1 \wedge x_2 \wedge x_3)$$

have "common factor" $(\widetilde{x}_1 \wedge x_2)$ i.e.

$$\begin{aligned} f &= (\widetilde{x}_1 \wedge x_2)(\widetilde{x}_3 \vee x_3) \vee (x_1 \wedge x_2 \wedge x_3) = \\ &= (\widetilde{x}_1 \wedge x_2) \vee (x_1 \wedge x_2 \wedge x_3) \end{aligned}$$

Unfortunately, there are no general rules how to find the "optimal" or "shortest" forms of Boolean polynomials (as well as in elementary algebra there are no algorithm for the factorization of polynomial functions).

We know now two different forms for Boolean functions: tables and Boolean polynomials (normal disjunctive forms). There is the third method for the same purposes: so-called circuit representation.

The circuit is a system of wires and gates (non-gates or inverters, and-gates and or-gates). The circuit has an input (containing a few wires) and an output.

Some of the wires can be split halfway and used as inputs for different gates. Each circuit represents a Boolean function and each Boolean function written as a polynomial in the normal form can be represented by circuit.

I'll illustrate the corresponding "transitions" from the circuit language to the Boolean functions language by examples having the complete information on the corresponding algorithms.

Construction of the Boolean function for a circuit.

The circuit is shown in Fig. 5.

Trace through the circuit from left to right indicating the outputs on each gate symbolically. See details on the fig 5.

The final expression is equal to

$$f(x_1, x_2, x_3) = (x_1 \vee x_2) \wedge [x_3 \vee \widetilde{(x_1} \wedge x_2)]$$

Using the DeMorgan law, one can transform the last factor:

$$[x_3 \vee \widetilde{(x_1} \wedge x_2)] = [\tilde{x}_3 \wedge (\tilde{x}_1 \vee \tilde{x}_2)]$$

The table of this function can be obtained by substituting the corresponding values from the left part of the table instead of variables x_1, x_2, x_3.

For instance,

$$\begin{aligned} f(0,0,0) &= (0 \vee 0) \wedge [\widetilde{0} \vee \widetilde{0})] = \\ &= 0 \wedge [1 \wedge 1] = 0 \wedge 1 = 0 \\ f(1,0,0) &= (1 \vee 0) \wedge [\widetilde{0} \wedge (\widetilde{1} \vee \widetilde{0})] = 1 \wedge [1 \wedge 1] = 1 \text{ etc.} \end{aligned}$$

The construction of the circuit for the given Boolean polynomial in the disjunctive form even simpler. First, construct monomials as shown below on specific example.

$$f(x_1, x_2, x_3) = x_1 \wedge \widetilde{x}_2 \wedge x_3$$

Construction of the disjunction is based on the following observation:

$$(f_1 \vee f_2 \vee f_3) = (f_1 \vee f_3) \vee f_3 = f_1 \vee (f_2 \vee f_3)$$

(associative law). It means that the circuits

and

give the same result and one can use the notation

The same since has the notation

using these observations and agreements one can represent, for instance, the Boolean function

$$f(x_1, x_2, x_3) = (x_1 \wedge x_2 \wedge \widetilde{x}_3) \vee (x_1 \wedge \widetilde{x}_2 \wedge x_3) \vee (\widetilde{x}_1 \wedge x_2 \wedge x_3)$$

by the circuit