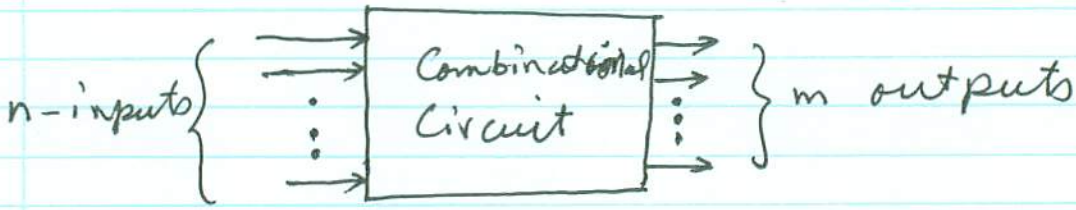
Lecture 8, Jan. 29, 2007

Analysis and Design of Combinational circuits

A Combinational circuit is a circuit whose output is a function of its inputs.

The Schematic for a Combinational circuit with n inputs and m outputs is:



A Combinational circuit can be described using a truth table with 2^n rows. Each output will generate a column having a 1 or a 0 for each of the 2^n input choices.

Analysis

The analysis is a procedure for finding the function that a given circuit performs.

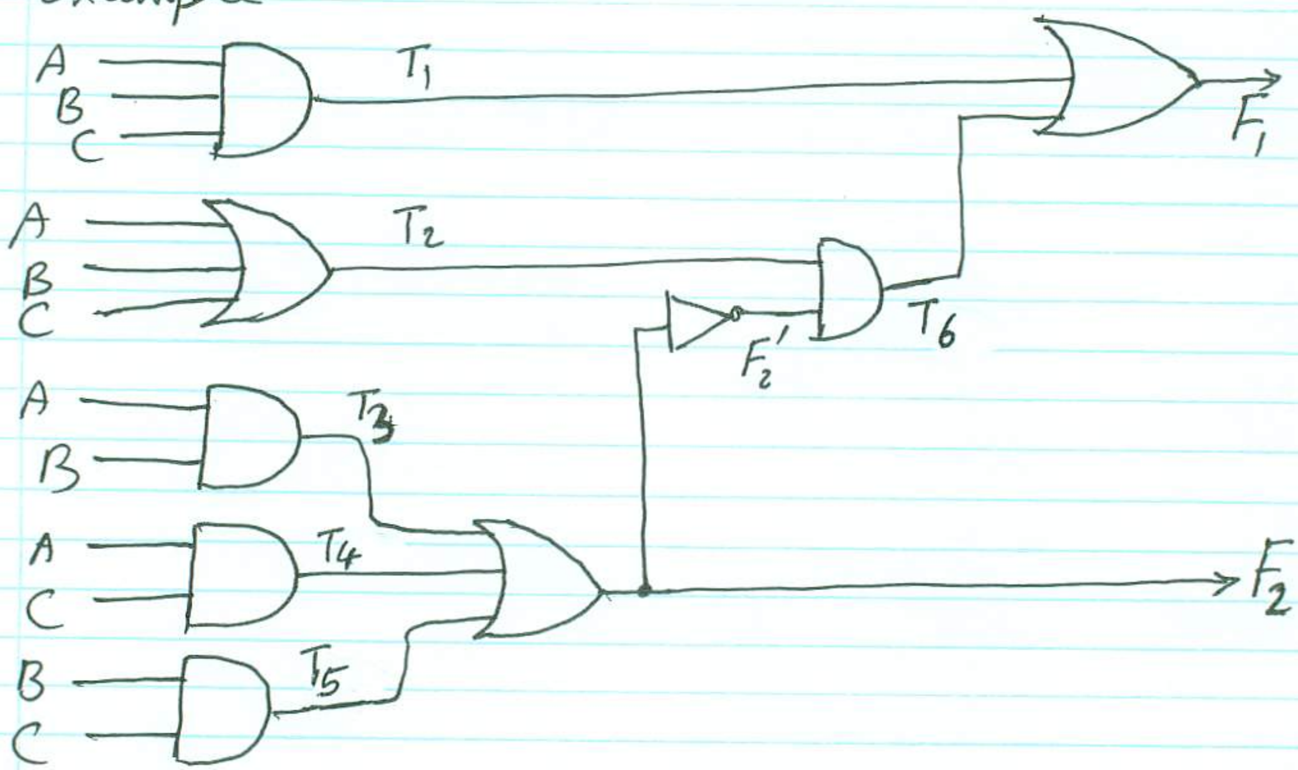
Starting from a circuit diagram, the analysis proceeds as follows:

1- Lable the outputs of the gate with relevant and distinct names.

2- Find the output of each gate in terms of its inputs starting from gates whose inputs are the inputs to the circuit and move on to those with inputs Consisting of the circuit inputs and outputs of other gates.

3- By substituting the outputs of the gates in early stages (those closer to the circuit's input) in later stage gates, derive the expression for the outputs of the circuit.

Example



Note that we have labeled the internal nodes with T_i, $i = 1, 2, \ldots, 6$ and the final nodes with F_1 and F_2.

$$T_1 = ABC$$

$$T_2 = A + B + C$$

$$T_3 = AB$$

$$T_4 = AC$$

$$T_5 = BC$$

$$F_2 = T_3 + T_4 + T_5 = AB + AC + BC$$

$$T_6 = T_2 F_2' = T_2 (AB + AC + BC)'$$
$$= (A + B + C)(AB + AC + BC)'$$

$$F_1 = T_1 + T_6 = ABC + (A + B + C)(AB + AC + BC)'$$
$$= ABC + (A+B+C)(A'+B')(A'+C')(B'+C')$$
$$= ABC + (AB' + A'B + A'C + B'C)(A'B' + C')$$
$$= ABC + AB'C' + A'BC' + A'B'C$$

So,

$$\boxed{F_1 = ABC + AB'C' + A'BC' + A'B'C}$$

and

$$\boxed{F_2 = AB + AC + BC}$$

While Boolean expressions are compact way of representing a circuit, they do not give insight into the operation of the circuit. To get more insight, we can draw the truth table from the Boolean expression. Of course, it is also possible to derive the truth table from the circuit diagram directly.

To derive the truth table follows the procedure below:

1) Make a table with 2^n rows where n is the number of inputs. Write numbers 0 to $2^n - 1$ on the rows of the table.

2- Label the outputs of the gates.

3- Form the truth table for those gates whose inputs are the inputs of the circuit.

4- Continue making truth table for other gates until you get to the outputs.

The truth table for the above example is shown below.

A	B	C	T_1	T_2	T_3	T_4	T_5	F_2	T_6	F_1
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	1	1
0	1	0	0	1	0	0	0	0	1	1
0	1	1	0	1	0	0	1	1	0	0
1	0	0	0	1	0	0	0	0	1	1
1	0	1	0	1	0	1	0	1	0	0
1	1	0	0	1	1	0	0	1	0	0
1	1	1	1	1	1	1	1	1	0	1

Note that F_1 is true, i.e., equal to 1 if there is either one or three 1's at the input and F_2 is equal to 1 if two or more 1's are among A, B and C. This is a full adder, a circuit that adds two bits and a carry and gives the sum of the three bits and an output carry.

Design

Design is the process of creating a combinational circuit from a functional specification. The design procedure consists of the following steps:

1- From the description of the circuit determine the number of the inputs and outputs. Assign a symbol to each input and each output.

2- Derive the truth table relating the inputs to each output.

3- Obtain a simplified Boolean function for each output.

4- Draw the logic diagram.

5- Verify the correctness of the design either manually or by simulation.

Example: Designing a Code Converter that has as its input 4 bits of a BCD (Binary Coded Decimal) value and outputs the Corresponding Excess-3 Code.

Binary Coded Decimal (BCD) is a way of representing digits 0 to 9 using natural 4-bit binary symbols 0000, 0001, ..., 1001.

Excess-3 Code represents digits 0 to 9 using 4 bits. But each number is formed by adding three (3) to the corresponding BCD codeword. For example 0 is represented as 0011, one is represented as 0100 ... and 9 is 1100. The interesting thing about Excess-3 Code is that if we invert each bit, we get the 9's complement of the codeword. For example, take 0110 that represent 3 and invert each bit to get 1001 which is the codeword for 6. Let's start with the truth table.

Input BCD Code				Excess-3 Code			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Then we draw the K-map for the four outputs w, x, y and z.

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

$z = D'$

AB \ CD	00	01	11	10
00	1		1	
01	1		1	
11	X	X	X	X
10	1		X	X

$y = CD + C'D'$

AB \ CD	00	01	11	10
00		1	1	1
01	1			
11	X	X	X	X
10		1	X	X

$x = B'C + B'D + BC'D'$

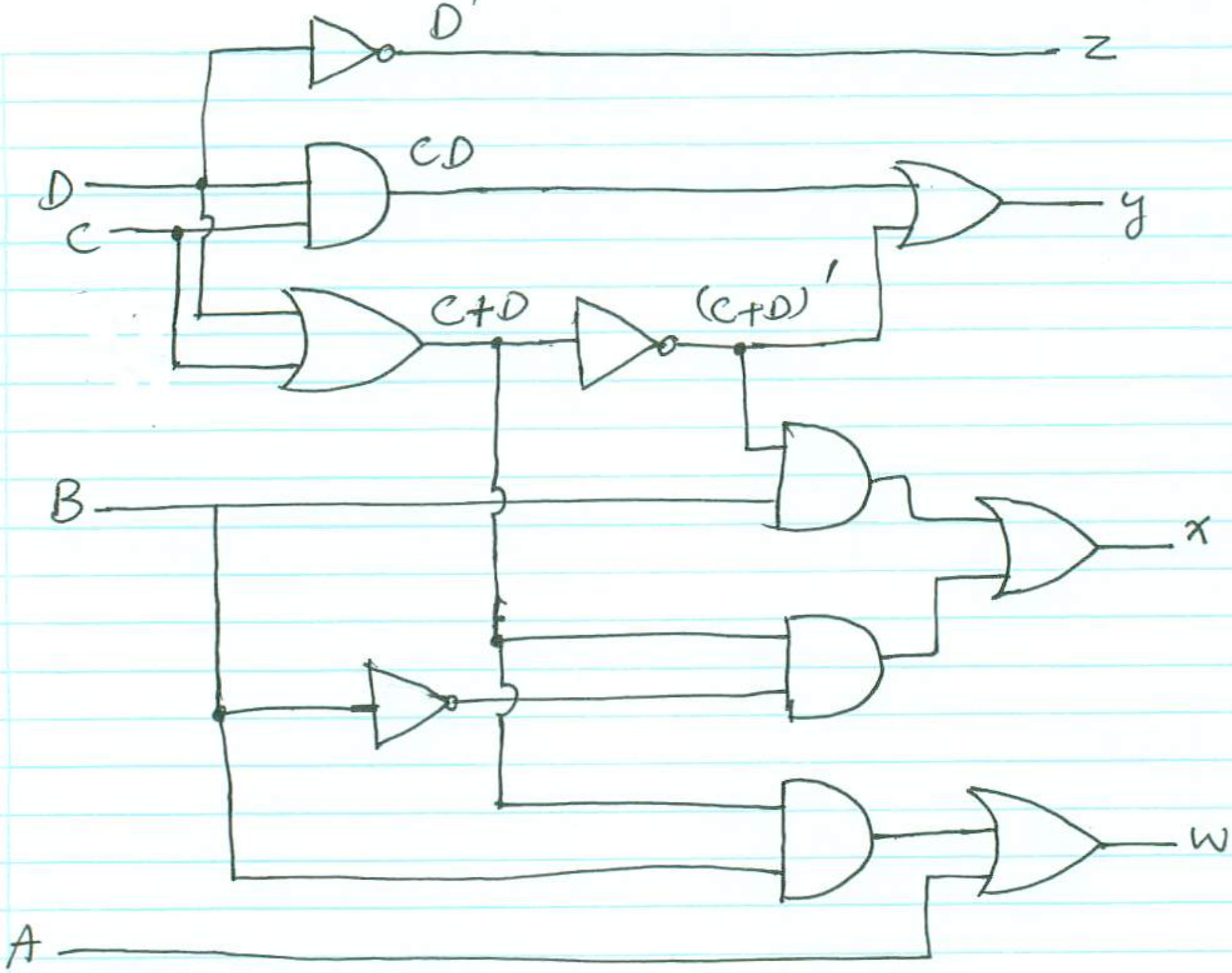
AB \ CD	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

$w = A + BD + BC$

So, we have

$$z = D'$$
$$y = CD + C'D' = CD + (C+D)'$$
$$x = B'C + B'D + BC'D' = B'(C+D) + B(C+D)'$$
$$w = A + BC + BD = A + B(C+D)$$

where we have manipulated the expression so that we can use similar two input gates. The implementation is show in the next page.



Digital circuit for BCD to Excess-3 Converter.