Combinational Logic: Combinational circuits, analysis procedure & design procedure

Lecture 1

Dronacharya group of institutions

Combinational v.s Sequential Circuits

Logic circuits may be combinational or sequential

Combinational circuits:

- Consist of **logic gates only.**
- Outputs are determined from the present values of inputs.
- Operations can be specified by a set of Boolean functions.

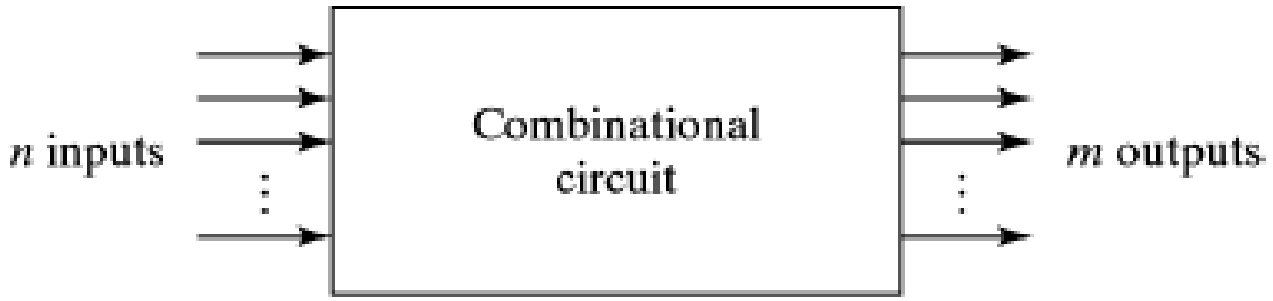
Sequential circuits:

- Consist of **logic gates and storage elements**
- Outputs are a function of the inputs and the state of the storage elements.
- Depend not only on present inputs, but also on past values
- Circuit behavior must be specified by a time sequence of inputs and internal states.

Combinational Circuit

A combinational circuit consists of

- Input variables
- Logic gates
- Output variables
- Block diagram of Combinational Circuit



Combinational Circuit

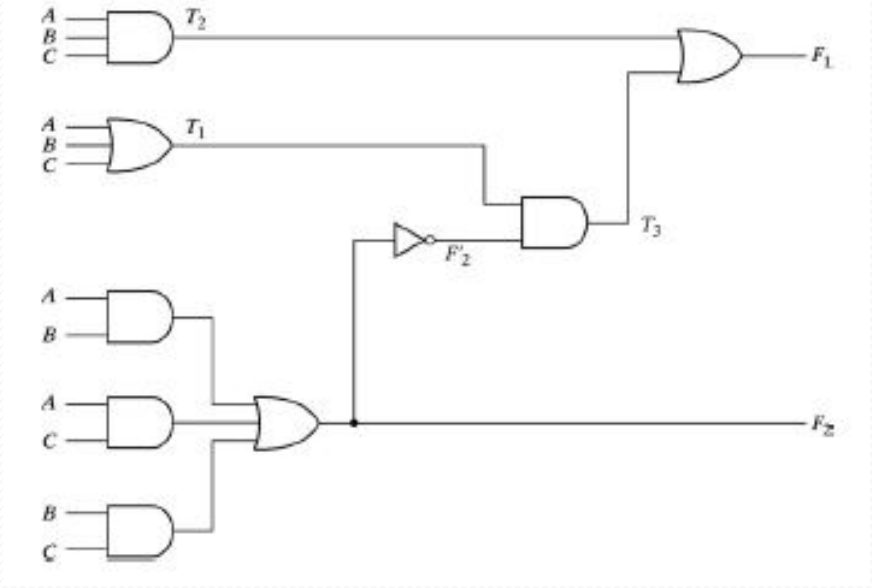
- Each input and output variable is a binary signal Represent logic 1 and logic 0.
- There are 2 n possible binary input combinations for n input variable.
- Only one possible output value for each possible input combination.
- Can be specified with a truth table.
- Can also be described by m Boolean functions, one for each output variable.
- Each output function is expressed in terms of n input variables.

Analysis Procedure

- Analysis: determine the function that the circuit implements
  - Often start with a given logic diagram
- The analysis can be performed by
  - Manually finding Boolean functions
  - Manually finding truth table
  - Using a computer simulation program
- First step: make sure that circuit is combinational
  - Without feedback paths or memory elements
- Second step: obtain the output Boolean functions or the truth table

Output Boolean Functions

- Step 1:
- Label all gate outputs that are a function of input variables.
- Determine Boolean functions for each gate output.



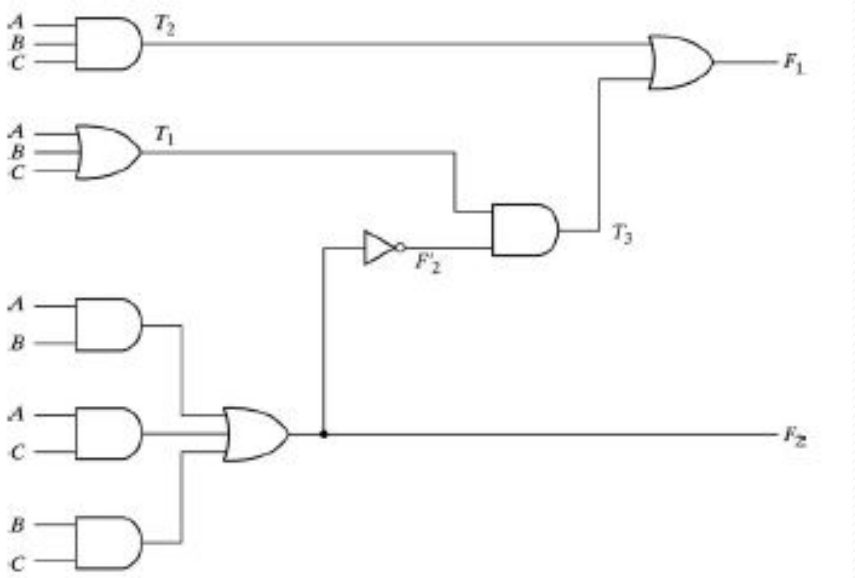
$F_2 = AB + AC + BC$

$T_1 = A + B + C$

$T_2 = ABC$

Output Boolean Functions

- Step 2:
- Label the gates that are a function of input variables and previously labeled gates.
- Find the Boolean function for these gates



$$T_3 = F'_2 T_1$$
$$F_1 = T_3 + T_2$$

Output Boolean Functions

- Step 3:
- Obtain the output Boolean function in term of input variables.
- By repeated substitution of previously defined functions.

$$
\begin{aligned}
F_1 &= T_3 + T_2 = F'_2\, T_1 + ABC \\
&= (AB + AC + BC)'\,(A + B + C) + ABC \\
&= (A' + B')(A' + C')(B' + C')\,(A + B + C) + ABC \\
&= (A' + B'\,C')(AB' + AC' + BC' + B'\,C) + ABC \\
&= A'\,BC' + A'\,B'\,C + AB'\,C' + ABC
\end{aligned}
$$

Truth Table

To obtain the truth table from the logic diagram:

1. Determine the number of input variables.

For n inputs:

- 2 n possible combinations
- List the binary numbers from 0 to 2 n-1 in a table

2. Label the outputs of selected gates.

3. Obtain the truth table for the outputs of those gates that are a function of the input variables only.

4. Obtain the truth table for those gates that are a function of previously defined variables at step 3.

- Repeatedly until all outputs are determined

Design Procedure

Design procedure:

- Input: the specification of the problem
- Output: the logic circuit diagram (or Boolean functions)

Step 1: determine the required number of inputs and outputs from the specification.

Step 2: derive the truth table that defines the required relationship between inputs and outputs.

Step 3: obtain the simplified Boolean function for each output as a function of the input variables.

Step 4: draw the logic diagram and verify the correctness of the design.

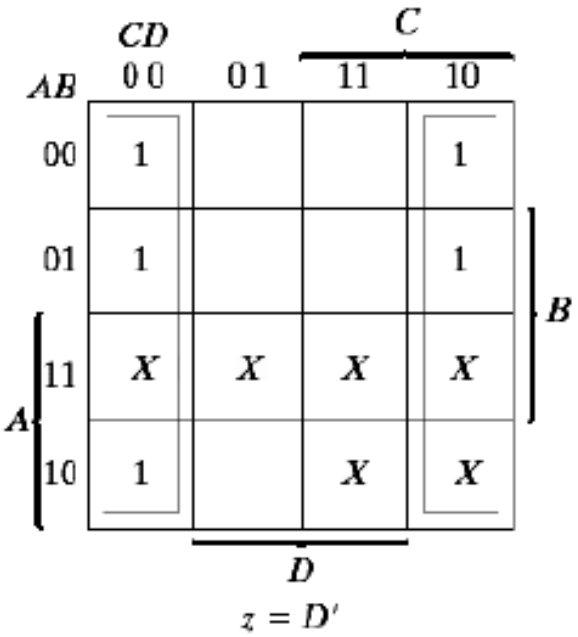
Code Conversion Example

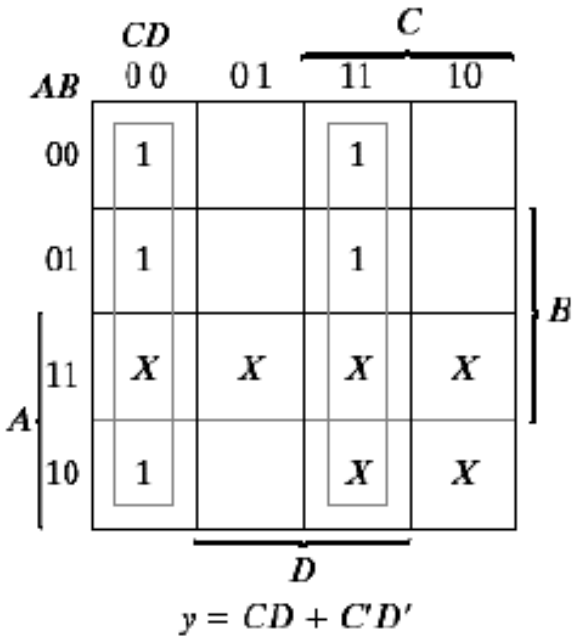
- Convert from BCD code to Excess-3 code.
- The 6 input combinations not listed are don't cares.
- These values have no meaning in BCD.
- We can arbitrary assign them to 1 or 0.

Input BCD				Output 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

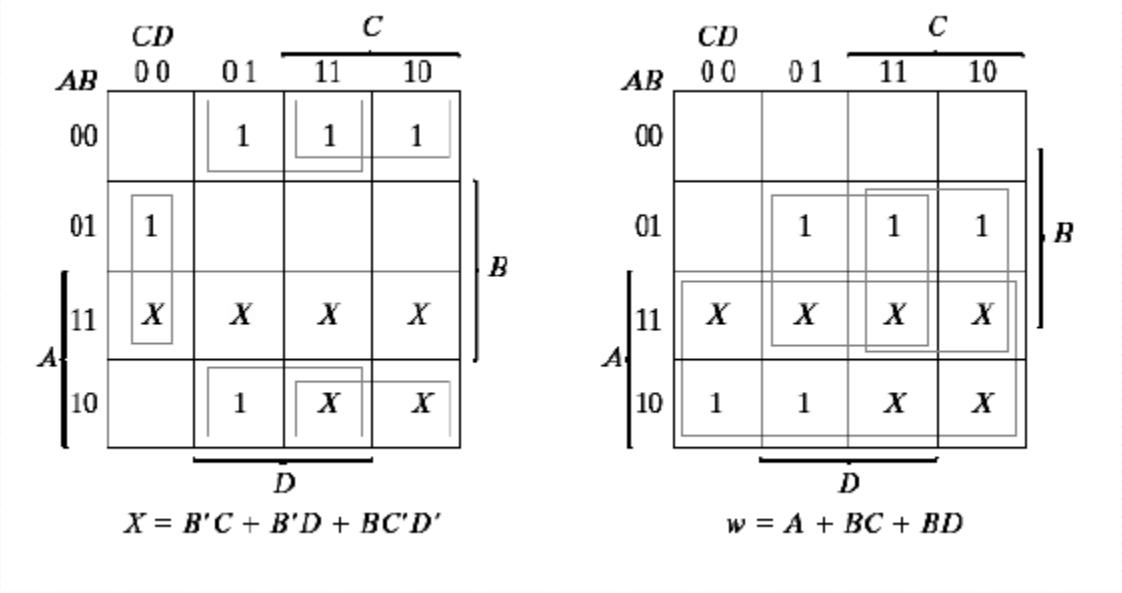
Maps for Code Converter

- The six don't care minterms (10~15) are marked with X.





Maps for Code Converter



Logic Diagram for the Converter

- There are various possibilities for a logic diagram that implements a circuit.
- A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps.
- The expressions may be manipulated algebraically to use common gates for two or more outputs.
- Reduce th

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

Logic Diagram for the Converter