Ondrej Harnúšek, ID: 79545
Tuesday 06:00 PM

ANALYSIS OF COMBINATIONAL CIRCUITS

Layout 9: NOR – NOR – NOR – XNOR – XOR

Make analysis of a combinational circuit, the structure of which is given in the image.

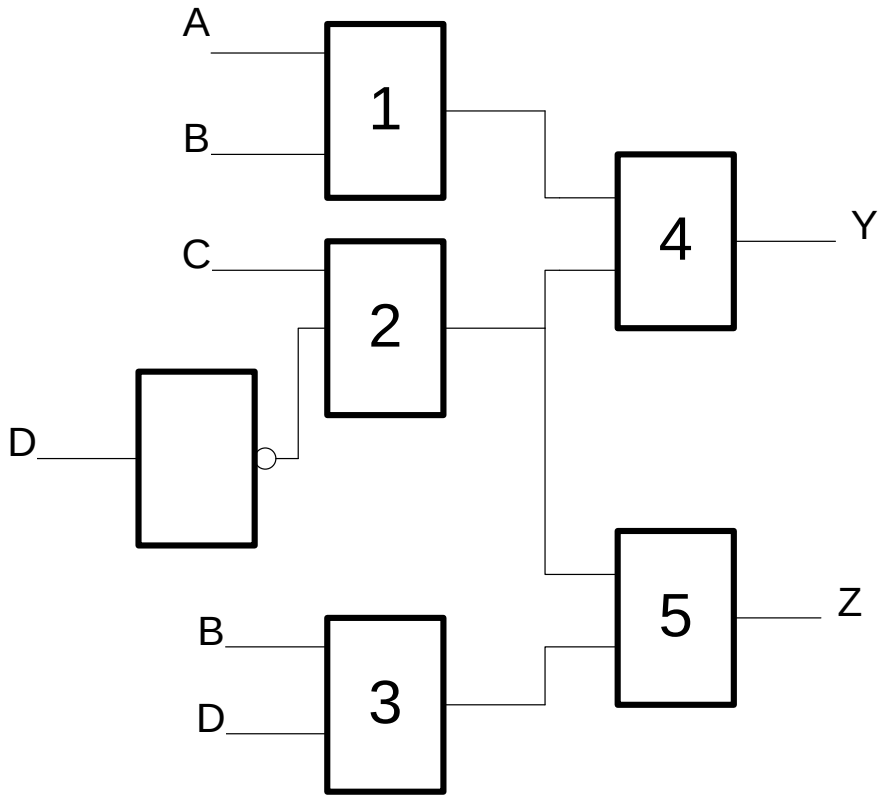
1. From a known circuit structure:

- Derive the Boolean functions corresponding with outputs Y and Z of circuit.
- Boolean functions using the rules of Boolean algebra change on the minimum DNF and write it to Karnaugh maps (firstly to maps, where are all input variables of circuit and then to the smallest maps).
- Boolean functions using the rules of Boolean algebra to change on the minimum KNF and write it to Karnaugh maps (firstly to maps, where are all input variables of circuit and then to the smallest maps).

2. Via system the LOGISIM (alt. LOG/FITBOARD):

- Create the schema a given circuit and through the simulation to verify the accuracy of map entries of Boolean functions (different combinations of the input values to compare with the outputs values in the maps).
- Create a schema of the equations, which were obtained when modification on the DNF.
- Create a schema of the equations, which were obtained when modification on the KNF.
- Connect properly all three schemes (use ramification from common inputs) and outputs of the circuits to place side by side (see. example of picture).

Schema of given circuit

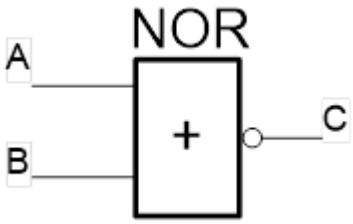


Ondrej Harnúšek, ID: 79545
Tuesday 06:00 PM

1) The types of used logic gates:NOR – NOR – NOR – XNOR – XOR

NOR Function

$$C = \overline{A + B}$$



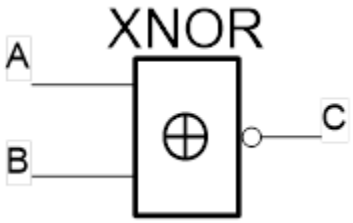
Schema

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

Table of truth values

XNOR Function

$$C = A.B + \overline{A}.\overline{B}$$



Schema

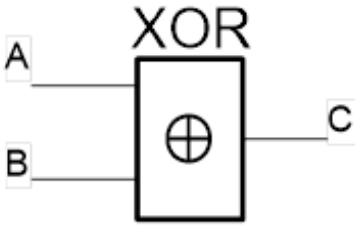
A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

Table of truth values

XOR Function

$$C = A.\overline{B} + \overline{A}.B$$

Ondrej Harnúšek, ID: 79545
Tuesday 06:00 PM



Schema

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Table of truth values

2) Based on the structure, we make the equations corresponding with outputs Y and Z:

$$Y = \overline{A+B}\,.\,\overline{C+\bar{D}} + \overline{\overline{A+B}}\,.\,\overline{\overline{C+\bar{D}}}$$
$$Z = \overline{C+\bar{D}}\,.\,\overline{\overline{B+D}} + \overline{\overline{C+\bar{D}}}\,.\overline{B+D}$$

3) We rewrite the equations to the equivalent normal form of type DNF:

Function Y:

$Y = 1.2 + \bar{1}.\bar{2}$

$1 = \overline{A+B}$ = — De Morgan's law

$= \bar{A}.\bar{B}$

$2 = \overline{C+\bar{D}}$ = — De Morgan's law

$= \bar{C}.\bar{\bar{D}}$ — Double negation

$= \bar{C}.D$

$Y = 1.2 + \bar{1}.\bar{2}$ — Substitutions of 1 and 2

$= (\bar{A}.\bar{B}).(\bar{C}.D) + (\overline{\bar{A}.\bar{B}}).(\overline{\bar{C}.D})$ — De Morgan's law

$= (\bar{A}.\bar{B}).(\bar{C}.D) + (\bar{\bar{A}}+\bar{\bar{B}}).(\bar{\bar{C}}+\bar{D})$ — Double negation

$= (\bar{A}.\bar{B}).(\bar{C}.D) + (A+B).(C+\bar{D})$ — Distributivity

$=(\bar{A}.\bar{B}.\bar{C}.D) + (A.C) + (A.\bar{D}) + (B.C) + (B.\bar{D})$

The number of used logic gates: 10 (4xNOT, 5xAND, 1xOR)
The number of inputs for logic function: 21 (1 to NOT, 1 to NOT, 1 to NOT, 1 to NOT, 4 to AND, 2 to AND, 2 to AND, 2 to AND, 2 to AND, 5 to OR)

Function Z:

$Z = 2.\bar{3} + \bar{2}.3$

$2 = \overline{C+\bar{D}}$ = — De Morgan's law

$= \bar{C}.\bar{\bar{D}}$ — Double negation

Ondrej Harnúšek, ID: 79545
Tuesday 06:00 PM

$= \bar{C}.D$	
$3 = \overline{B+D} =$	De Morgan's law
$= \bar{B}.\bar{D}$	
$Z = 2.\bar{3} + \bar{2}.3$	Substitutions of 2 and 3
$= (\bar{C}.D).(\overline{\bar{B}.\bar{D}}) + (\overline{\bar{C}.D}).(\bar{B}.\bar{D})$	De Morgan's law
$= (\bar{C}.D).(\bar{\bar{B}} + \bar{\bar{D}}) + (\bar{\bar{C}} + \bar{D}).(\bar{B}.\bar{D})$	Double negation
$= (\bar{C}.D).(B+D) + (C+\bar{D}).(\bar{B}.\bar{D})$	Distributivity
$= (\bar{C}.B + \bar{C}.D).(D.B + D.D) + (\bar{B}.C + \bar{B}.\bar{D}).(\bar{D}.C + \bar{D}.\bar{D})$	Multiple equal inputs
$= (\bar{C}.B + \bar{C}.D).(D.B + D) + (\bar{B}.C + \bar{B}.\bar{D}).(\bar{D}.C + \bar{D})$	Absorption
$= (\bar{C}.B + \bar{C}.D).(D) + (\bar{B}.C + \bar{B}.\bar{D}).(\bar{D})$	Distributivity
$= (\bar{C}.B.D + \bar{C}.D.D) + (\bar{B}.C.\bar{D} + \bar{B}.\bar{D}.\bar{D})$	Multiple equal inputs
$= (\bar{C}.B.D + \bar{C}.D) + (\bar{B}.C.\bar{D} + \bar{B}.\bar{D})$	Distributivity
$= (\bar{C}.D).(B+1) + (\bar{B}.\bar{D}).(C+1)$	Annihilator for OR
$= (\bar{C}.D).1 + (\bar{B}.\bar{D}).1$	
$= \bar{C}.D + \bar{B}.\bar{D}$	

The number of used logic gates: 6 (3xNOT, 2xAND, 1xOR)
The number of inputs for logic function: 9 (1 to NOT, 1 to NOT, 1 to NOT, 2 to AND, 2 to AND, 2 to OR)

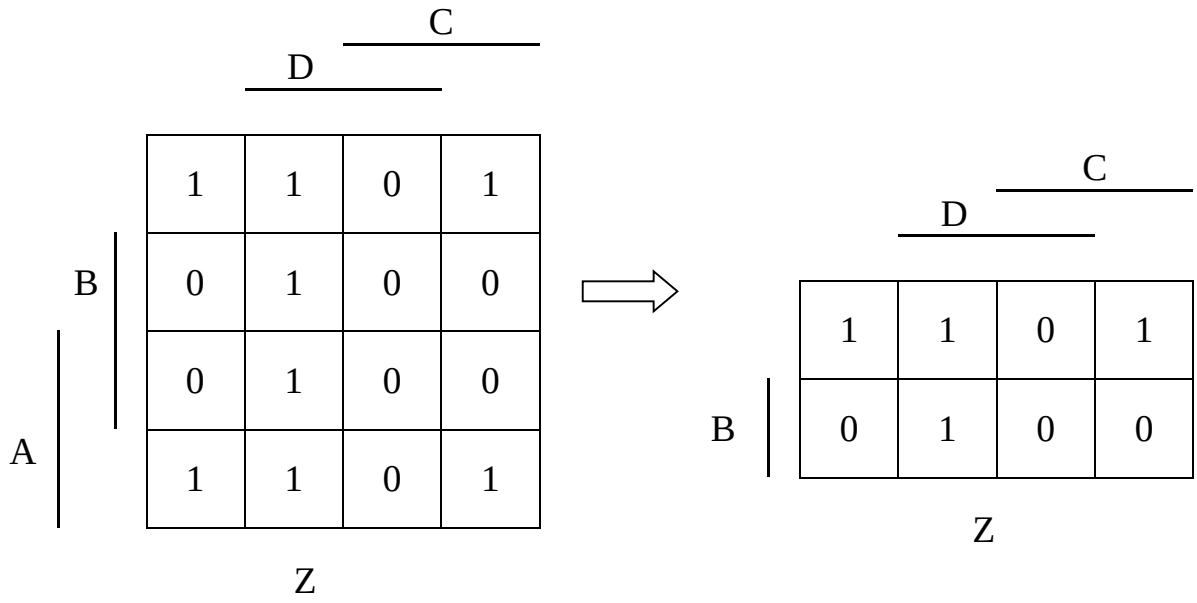
Summary of circuit:
The number of used logic gates: 13 (4xNOT, 7xAND, 2xOR)
The number of inputs for logic function: 24 (1 to NOT, 1 to NOT, 1 to NOT, 1 to NOT, 4 to AND, 2 to AND, 2 to AND, 2 to AND, 2 to AND, 2 to AND, 2 to OR, 2 to OR)

4) We compile maps of functions which correspond to expressions Y and Z:

A \ B	CD = 00	CD = 01	CD = 11	CD = 10
00	0	1	0	0
01	1	0	1	1
11	1	0	1	1
10	1	0	1	1

Y

Ondrej Harnúšek, ID: 79545
Tuesday 06:00 PM



5) We rewrite the equations to the equivalent normal form of type KNF:

$Y = (\bar{A}.\bar{B}.\bar{C}.D) + (A.C) + (A.\bar{D}) + (B.C) + (B.\bar{D})$ Distributivity: (A) and (B)

$= (\bar{A}.\bar{B}.\bar{C}.D) + A.(C + \bar{D}) + B.(C + \bar{D})$ Distributivity: $(C + \bar{D})$

$= (\bar{A}.\bar{B}.\bar{C}.D) + (A + B).(C + \bar{D})$ Rule 3a: $\underline{\boldsymbol{A}}=(\bar{A}.\bar{B}.\bar{C}.D)$, $\underline{\boldsymbol{B}}=(A + B)$,$\underline{\boldsymbol{C}}=(C + \bar{D})$

$= \big((\bar{A}.\bar{B}.\bar{C}.D) + (A + B)\big).((\bar{A}.\bar{B}.\bar{C}.D) + (C + \bar{D}))$ Removing brackets

$= ((\bar{A}.\bar{B}.\bar{C}.D) + A + B).\ ((\bar{A}.\bar{B}.\bar{C}.D) + C + \bar{D})$ Rule 12a

$= ((\bar{C}.D) + A + B).\ ((\bar{A}.\bar{B}) + C + \bar{D})$ Rule 3a. left side:$\underline{\boldsymbol{A}}=(A + B)$, *right side:* $\underline{\boldsymbol{A}}=(C + \bar{D})$

$= (A + B + \bar{C}).(A + B + D).(\bar{A} + C + \bar{D}).(\bar{B} + C + \bar{D})$

The number of used logic gates: 7 (4xNOT, 1xAND, 4xOR)
The number of inputs for logic function: 20 (1 to NOT, 1 to NOT, 1 to NOT, 1 to NOT, 3 to OR, 3 to OR, 3 to OR, 3 to OR, 4 to AND)

$Z = \bar{C}.D + \bar{B}.\bar{D}$ Rule 3a ($\underline{\mathbf{A}}=\bar{C}.D$)

$= (\bar{C}.D + \bar{B}).(\bar{C}.D + \bar{D})$ For first brackets we use Rule 3a

$= (\bar{C} + \bar{B}).(D + \bar{B}).(\bar{C}.D + \bar{D})$ For second brackets we use Rule 3a

$= (\bar{C} + \bar{B}).(D + \bar{B}).(\bar{C} + \bar{D}).(\bar{D} + D)$ For last brackets we use Rule 7a

$= (\bar{C} + \bar{B}).(D + \bar{B}).(\bar{C} + \bar{D}).(1)$ For first brackets we use Rule 9b

$= (\bar{B} + \bar{C}).(\bar{B} + D).(\bar{C} + \bar{D})$ Consensus theorem

$= (\bar{B} + D).(\bar{C} + \bar{D})$

The number of used logic gates: 6 (3xNOT, 1xAND, 2xOR)
The number of inputs for logic function: 9 (1 to NOT, 1 to NOT, 1 to NOT, 2 to OR, 2 to OR, 2 to AND)

Summary of circuit:
The number of used logic gates: 8 (4xNOT, 2xAND, 6xOR)
The number of inputs for logic function: 26 (1 to NOT, 1 to NOT, 1 to NOT, 1 to NOT, 3 to OR, 3 to OR, 3 to OR, 3 to OR, 2 to OR, 2 to OR, 4 to AND, 2 to AND)

Ondrej Harnúšek, ID: 79545
Tuesday 06:00 PM

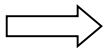
6) We compile maps of functions which correspond to expressions Y and Z:

AB \ CD	00	01	11	10
00	0	1	0	0
01	1	0	1	1
10	1	0	1	1
11	1	0	1	1

Y

AB \ CD	00	01	11	10
00	1	1	0	1
01	0	1	0	0
10	0	1	0	0
11	1	1	0	1

Z



B \ CD	00	01	11	10
0	1	1	0	1
1	0	1	0	0

Z

Ondrej Harnúšek, ID: 79545
Tuesday 06:00 PM

Final assessment:

The first task was to derive the Boolean functions from known structure of the combinational logic circuit, which corresponding with outputs Y and Z. Then, to change functions to the minimum DNF, using the rules of Boolean algebra (mostly De Morgan's law and law about double negation). Such modified functions we had to write to the Karnaugh maps. On map for output "Z" may be seen that the variable "A" does not affect the output, so the map can be further reduced.

Similarly, the functions could be adjusted to a minimum DNF, according to the rules: distributivity (we used the substitution), rule about completeness or consensus theorem. This form of functions was entered to the Karnaugh maps and map for output "Z" could still be simplified by removing the variable "A".

Correctness of changed functions mDNF mKNF can be verified from their Karnaugh maps, which are identical with maps functions of a given combinational logic circuit.

Another task was to create a schemas for a given circuit, the circuit of equation DNF, the circuit of equation mKNF and all three circuits to connect on common inputs. By simulation we can verify the identity of outputs of all three circuits and values in the maps.

From the schema is clear that it is more convenient to use for the creation of circuit the equations KNF (whereby the logic circuit has 8 gates) as DNF (where it is needed 13 logic gates).

Ondrej Harnúšek, ID: 79545
Tuesday 06:00 PM