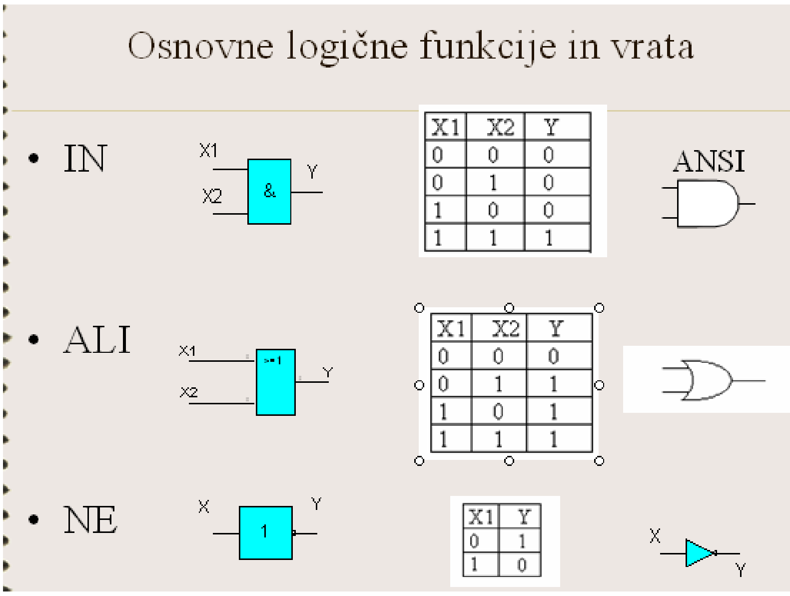
Osnovne logične funkcije

Poznamo tri osnovne logične funkcije: IN ali logično konjunkcijo, ALI ali logično disjunkcijo in NE ali negacijo.

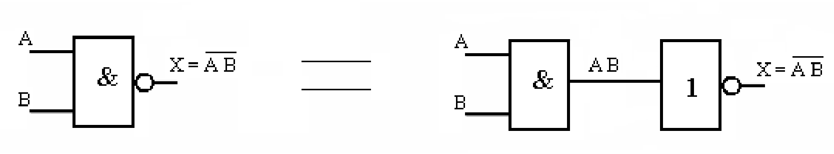


Slika 1: Osnovne logične funkcije

Sestavljene logične funkcije

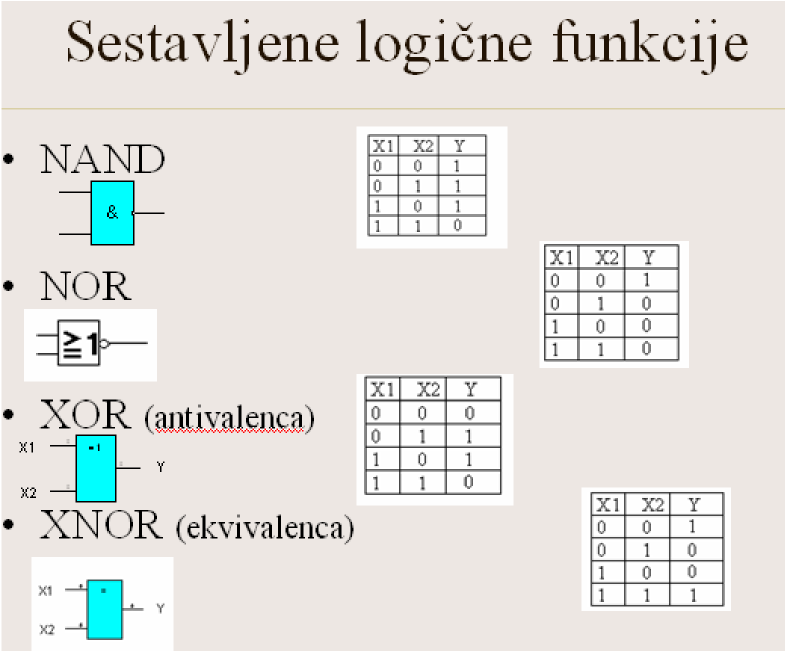
Logične funkcije sestavljamo iz osnovnih funkcij. Nekatere so pogostejše in jih dobimo v integrirani izvedbi. Te standardne:

NE-IN ali negirani IN ali **NAND**, ki je kombinacija IN in negatorja



A	B	A B	X
0	**0**	0	**1**
1	**0**	0	**1**
0	**1**	0	**1**
1	**1**	1	**0**

Slika 2: Simbol, vezje in logična tabela za NE-IN



Slika 3: Pregled standardnih sestavljenih logičnih funkcij

Zapisovanje logičnih funkcij

Poznamo več načinov zapisovanja logičnih funkcij oziroma enačb. Mi si bomo pogledali le najbolj pogoste načine.

Logična tabela
To je tabela, ki ima določeno število vhodov in izhodov. V tem primeru imamo dva vhoda, označena z A in B, ter en izhod, označen s črko X. Vhoda A in B nam predstavljata dve stikali ali dva senzorja, izhod pa žarnico ali rele, preko katerega vklopimo npr. motor, grelec, elektromagnet,.. Logična 1 na izhodu pomeni, da žarnica sveti, 0 pa, da ne sveti.

Za vhoda napišemo vse možne kombinacije. Ker sta vhoda dva, je število možnih kombinacij 2^2=4. (Če bi bili trije vhodi, bi imeli osem možnih kombinacij na vhodu, ker je 2^3=8.)

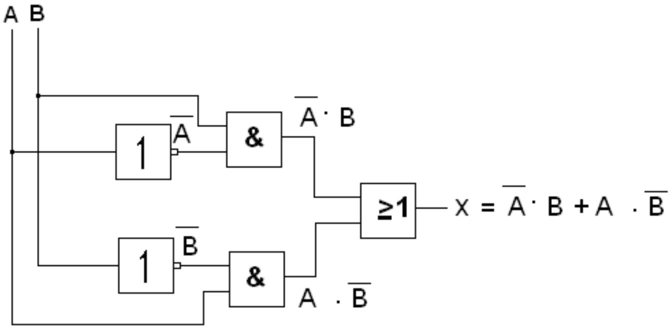
A	B	X	
0	**0**	**0**	Obe stikali sta izključeni – žarnica ne sveti.
0	**1**	**1**	Sklenjeno je samo stikalo A – žarnica sveti - $\overline{A} \cdot B$
1	**0**	**1**	Sklenjeno je samo stikalo B – žarnica sveti - $A \cdot \overline{B}$
1	**1**	**0**	Obe stikali sta sklenjeni – žarnica ne sveti.

Naša funkcija ima vrednost 1, ko je sklenjeno eno izmed dveh stikal. To zapišemo z logično enačbo

$$X = \overline{A} \cdot B + A \cdot \overline{B}$$

Žarnica sveti, ko je sklenjeno stikalo A, medtem ko stikalo B ni sklenjeno, pa tudi takrat, ko je sklenjeno stikalo A, B pa ne. Ker ima izhod X dve enici, sta ti povezani s funkcijo ALI.

Iz enačbe narišemo logično vezje ali funkcijski načrt:



Slika 4: Funkcijski načrt za logično funkcijo XOR.

Z logičnimi funkcijami opišemo tudi delovanje posameznih digitalnih vezij in naprav. Podajamo jih na različne načine, najpogosteje z logičnimi tabelami, enačbami ali funkcijskim načrtom.

V spodnji tabeli je zapisana funkcija X. V obliki enačbe jo zapišemo tako, da s funkcijo ALI povežemo vse tiste člene, pri katerih ima funkcija vrednost **1**. Tem členom s tujko rečemo **mintermi.** Desno od tabele so napisani vsi členi, ne glede na njihovo vrednost, potem pa jih bomo pomnožili z 0 ali 1, odvisno od tega, kakšno vrednost ima pri tem funkcija.

A	B	C	X	
0	0	0	0	m_0
0	0	1	1	m_1
0	1	0	0	m_2
0	1	1	1	m_3
1	0	0	0	m_4
1	0	1	0	m_5
1	1	0	1	m_6
1	1	1	1	m_7

$m_0 = \overline{A}\ \overline{B}\ \overline{C}$ $\quad$ $m_4 = A\ \overline{B}\ \overline{C}$

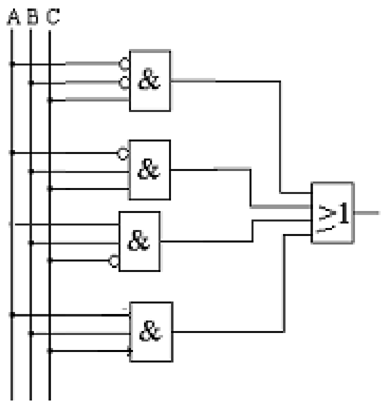
$m_1 = \overline{A}\ \overline{B}\ C$ $\quad$ $m_5 = A\ \overline{B}\ C$

$m_2 = \overline{A}\ B\ \overline{C}$ $\quad$ $m_6 = A\ B\ \overline{C}$

$m_3 = \overline{A}\ B\ C$ $\quad$ $m_7 = A\ B\ C$

Enačba za zgornjo logično funkcijo je: $X = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C + A \cdot B \cdot \overline{C} + A \cdot B \cdot C$

Logično vezje po spodnji enačbi je:



Slika 5: Logično vezje

Krajšanje logičnih funkcij

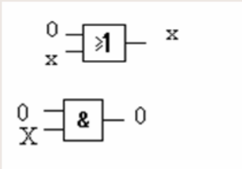
Marsikatero logično funkcijo lahko poenostavimo – narišemo jo z veliko manj elementi.
To lahko naredimo s pomočjo algebre dvo-vrednostne logike. Za logični ALI lahko uporabimo znak + ali v, za IN pa & ali · ali *.

Poglejmo nekaj matematičnih pravil, ki veljajo v dvo-vrednostni logiki oz. v Boolovi algebri:

Boolova algebra

X v 0 = X + 0 = X

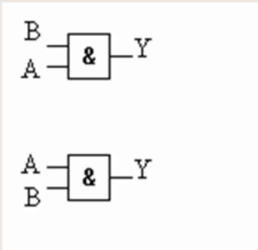
X * 0 = 0



Komutativost ali zamenljivost –

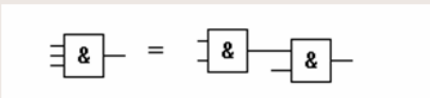
A + B = B + A

A * B = B * A



Asociativnost ali združljivost

(A+B)+C = A+(B+C) = A + B + C



Distributivnost ali razdružljivost

A * (B v C) = (A * B) v (A * C)

A v (B * C) = (A v B) * (A v C)

$X + \overline{X} = 1$

X + 1 = 1

IDEMPOTENCA

X + X + X = X **X * X *X = X**

INVOLUCIJA

$\overline{\overline{X}} = X$

ABSORBCIJA

X + XY = X **$(X + \overline{Y}) * Y = XY$**

X * (X + Y) = X **$X\overline{Y} + Y = X + Y$**

De Morganov izrek

$\overline{X + Y} = \overline{X} \cdot \overline{Y}$ in

$\overline{X \cdot Y} = \overline{X} + \overline{Y}$

Vrstni red operacij:

1. Negacija
2. Konjunkcija
3. Disjunkcija

S pomočjo DeMorganovega izreka lahko iz negatorjev in IN vrat naredimo ALI vrata in obratno, iz ALI vrt in negatorjev IN vrata. Večina digitalnih vezij je narejenih samo z NAND vrati.

Z uporabo pravil **Boolove algebre** lahko okrajšamo logično funkcijo tako, da izpostavimo skupne faktorje:

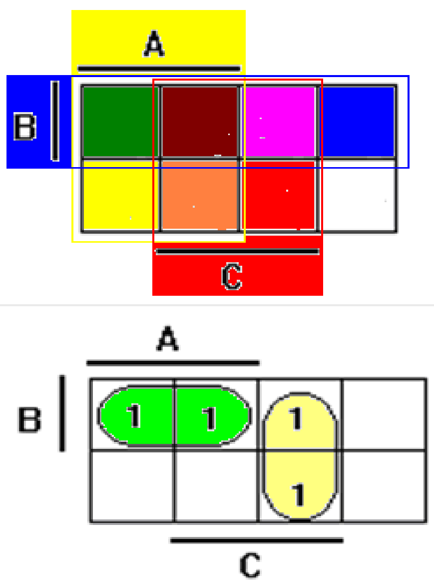
$$X = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C + A \cdot B \cdot \overline{C} + A \cdot B \cdot C = \overline{A} \cdot C(\overline{B} + B) + A \cdot B(\overline{C} + C) = \overline{A} \cdot C + A \cdot B$$

Izraza v oklepajih imata vrednost 1 in dobimo končni rezultat $X = \overline{A} \cdot C + A \cdot B$

Do enake rešitve pridemo z grafično metodo, s tako imenovanim VK (Veitch – Caurnaught) diagramom.

Veitchev ali VK diagram

V pravokotniku imamo toliko prostorčkov, kolikor je lahko največ vrstic v tabeli (za tri vhodne spremenljivke osem). Vhodne spremenljivke razporedimo tako, da vsaka pokriva natanko polovico celotnega prostora. Vedno moramo imeti en prostorček, ki ga ne pokriva nobena spremenljivka in prostorček, ki ga prekrivajo vse spremenljivke. V prostorčke vpisujemo enice. Nato sosede z vrednostjo 1 združimo (sosedi so levo, desno, spodaj, zgoraj) in napišemo skupne lastnosti.

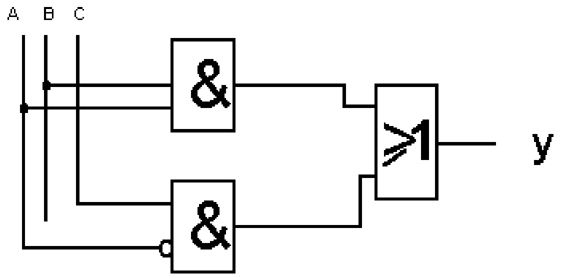


A	B	C	X	
0	0	0	0	$\overline{A}\overline{B}\overline{C}$
0	0	1	1	$\overline{A}\overline{B}C$
0	1	0	0	$\overline{A}B\overline{C}$
0	1	1	1	$\overline{A}BC$
1	0	0	0	$A\overline{B}\overline{C}$
1	0	1	0	$A\overline{B}C$
1	1	0	1	$AB\overline{C}$
1	1	1	1	ABC

$$Y = AB + \overline{A}C$$

Slika 6: Primer krajšanja logične funkcije z Veitch-evim diagramom.

Narišimo še logično vezje ali funkcijski načrt:



Slika 7: Funkcijski načrt okrajšane logične funkcije

Obe vezji (s slike 5 in slike 7) opravljata popolnoma isto logično funkcijo, le da je vezje na sliki 7 veliko manjše.

Primeri logičnih funkcij

Primer 1: **Alarmna naprava**
Naredimo enostavno alarmno napravo, ki ima dva svetlobna senzorja, in stikalo za vklop naprave. Svetlobni senzor ima osnovno stanje 1, ko pa prekinemo svetlobni žarek, gre izhod senzorja v stanje 0 in to sproži alarm. Alarm sproži katerikoli senzor, seveda le takrat, ko je naprava vključena (stikalo).

1. Določimo število vhodov in izhodov:
   Vhodi so trije:
   Stikalo S0
   Svetlobni senzor S1
   Svetlobni senzor S2

Naredimo prireditveni seznam ali tabelo, kamor zapišemo, kam priključimo vhode in kaj pomenijo:

Oznaka v načrtu		Naslov (pri krmilnikih)	Vrsta kontakta	Pomen	Opomba
S0	A		stikalo	1 – naprava vključena	
S1	B		Svetlobno stikalo	0 – ko prekinemo svetlobni tok	
S2	C		Svetlobno stikalo	0 – ko prekinemo svetlobni tok	

Izhod je le eden:

Oznaka v načrtu	Naslov (pri krmilnikih)	Aktiven pri	Pomen	Opomba
Y		1	1 – vklopi alarm	

2. Napišemo logično tabelo:

A(S0)	B(S1)	C(S2)	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

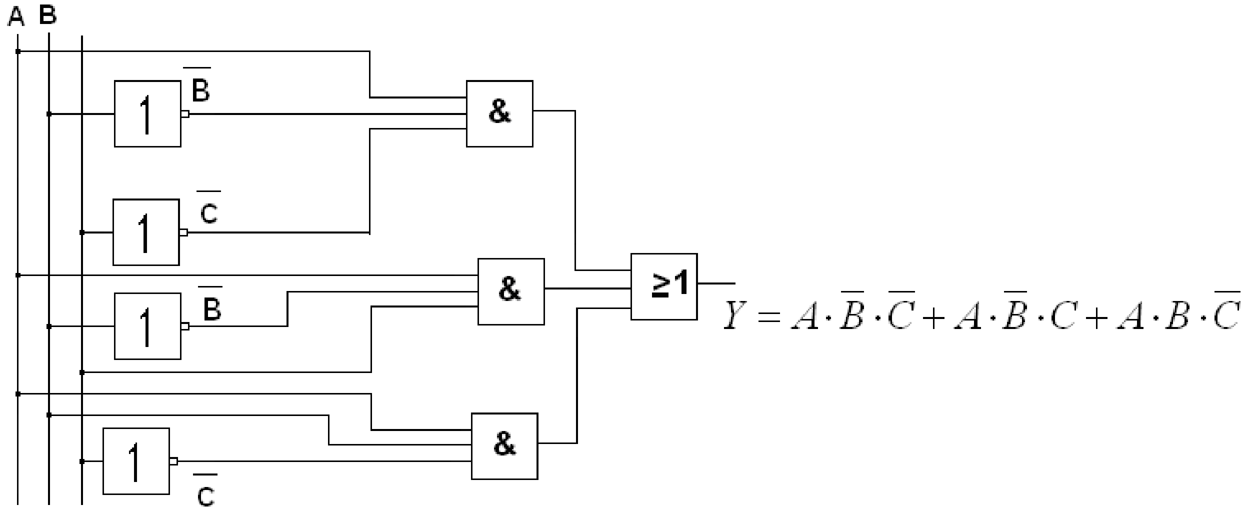
Stikalo je na 0 – naprava izključena.

Oba senzorja sta aktivirana.
Aktiviran je senzor S1.
Aktiviran je senzor S2.
Naprava je vključena, senzorja nista aktivirana – ni alarma.

3. Iz tabele napišemo enačbo:

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

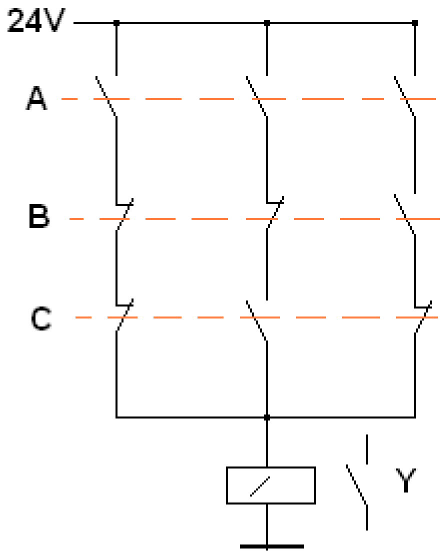
4. Logično funkcijo narišemo z logičnimi vrati (**funkcijski** načrt) ali s kontakti relejev oz. s stikali (**krmilni** načrt).



Slika 8: Funkcijski načrt

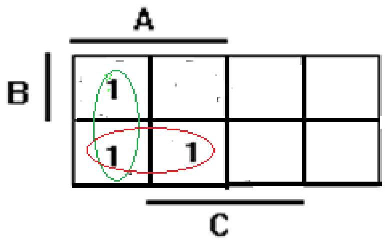
Krmilni načrt ali realizacija logične funkcije s stikali
Negirano spremenljivko predstavimo kot sklenjeno stikalo, nenegirano pa kot odprto stikalo. Zaporedno vezana stikala predstavljajo logični IN, vzporedna pa logični ALI.

S t.i. LADDER diagrami ali kontaktnimi načrti, ki so zelo podobni krmilnemu načrti, programiramo nekatere prosto programirljive krmilnike (npr. Siemensove krmilnike).



Slika 9: Krmilni načrt za zgornjo logično funkcijo

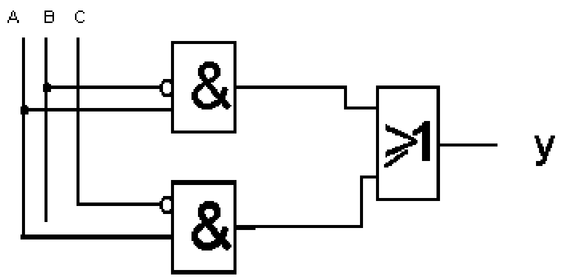
5. Krajšanje



Okrajšana logična funkcija je:

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

Vezje iz okrajšane enačbe:



Primer 2: **Prekodirnik iz BCD koda v kod za 7-segmentni prikazovalnik**

BCD kod je najbolj enostaven način, da neko število zapišemo z ničlami in enicami. Vsako desetiško cifro zapišemo v dvojiškem številskem sistemu na štirih mestih:

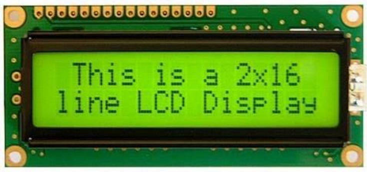
Primeri:

Desetiško število	Število v BCD kodu
17905	0001 0111 1001 0000 0101
1004	0001 0000 0000 0100
95	1001 0101

Prikazovalnik ali display je izhodna enota.
Lahko so:

- Numerični (izpišejo le številke in nekatere črke)
    - LED (velika poraba energije, dobra vidljivost)

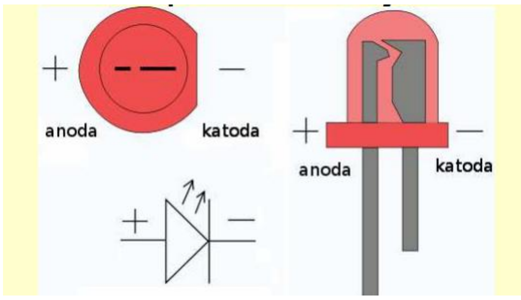
    - LCD (majhna poraba, slab vidljivost)
- Alfanumerični (številke, črke in posebne znake)
    - Največkrat so to LCD prikazovlniki, ki se med seboj razlikujejo po številu vrstic in mest v vrstici (npr. 2x16 ali 4x20)



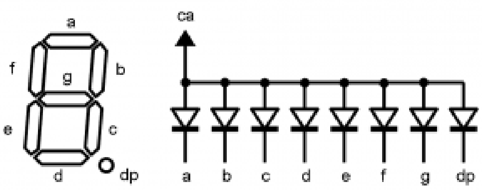
- Grafični (črke, številke, grafi), ki so LED, LDC, OLED…

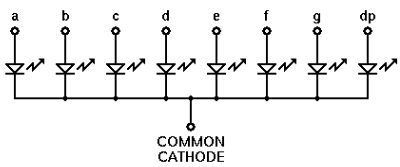
LED 7-segmentni prikazovalniki

Prikazovalnik je sestavljen iz svetlečih diod (LED). Vsaka dioda je en segment.
LED je elektronski element, ki oddaja svetlobo, ko skoznjo teče električni tok (4 do 20mA). Tok pa teče, če je na pozitivni elektrodi (anodi) + napetost, na negativni elektrodi (katodi) pa masa ali negativna napetost. Tok moramo omejiti z uporom.



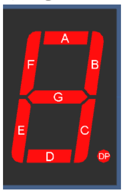
Poznamo dve vrsti LED prikazovalnikov:
s skupno anodo(CA) in s skupno katodo (CC).





Načrtovanje prekodirnika:

Najprej bomo sestavili logično tabelo. Imamo 4 vhode (ABCD, ker je BCD kod 4-biten) in 7 izhodov (segmenti a,b,c,d,e,f,g).



Slika 10: 7-segmentni prikazovalnik

Če je prikazovalnik zgrajen s skupno katodo, bo LED dioda zasvetila takrat, ko bo na segmentu prikazovalnika logična »1«. Ker gre za BCD kodo, se kombinacije od 1010 do 1111 ne bodo pojavile, zato lahko te kombinacije označimo kot redundantne z »x«. Koristile nam bodo pri minimizaciji posameznih izhodov.

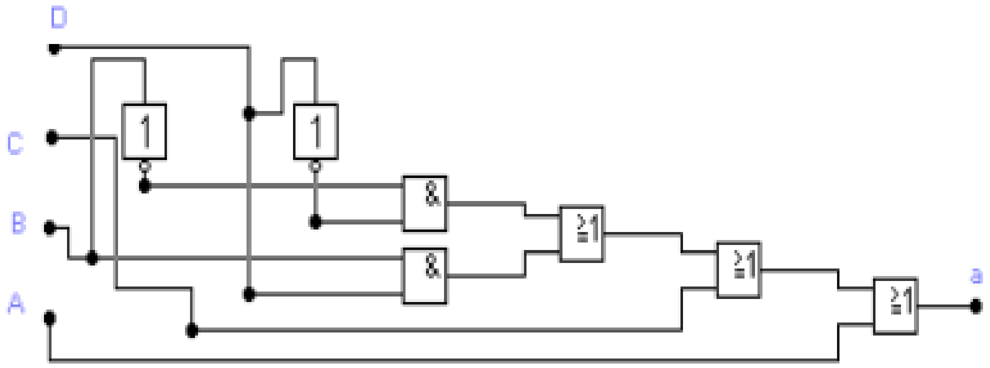
A	B	C	D	a	b	c	d	e	f	g
0	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**0**
0	0	0	1	**0**	1	1	0	0	0	0
0	0	1	0	**1**	1	0	1	1	0	1
0	0	1	1	**1**	1	1	1	0	0	1
0	1	0	0	**0**	1	1	0	0	1	1
0	1	0	1	**1**	0	1	1	0	1	1
0	1	1	0	**1**	0	1	1	1	1	1
0	1	1	1	**1**	1	1	0	0	0	0
0	0	0	0	**1**	1	1	1	1	1	1
1	0	0	1	**1**	1	1	1	0	1	1
1	0	1	0	**x**	x	x	x	x	x	x
.	.	.	.	**x**	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

Če damo na vhod BCD kod števila 0, mora display prikazati številko nič. Svetijo vsi segmenti razen segmenta g.

V tabeli imamo 7 logičnih funkcij. Okrajšajmo in narišimo vezje samo za segment a.

	A				
B	x	x	1		
	x	x	1	1	D
	1	x	1		
	1	x	1	1	
		C			

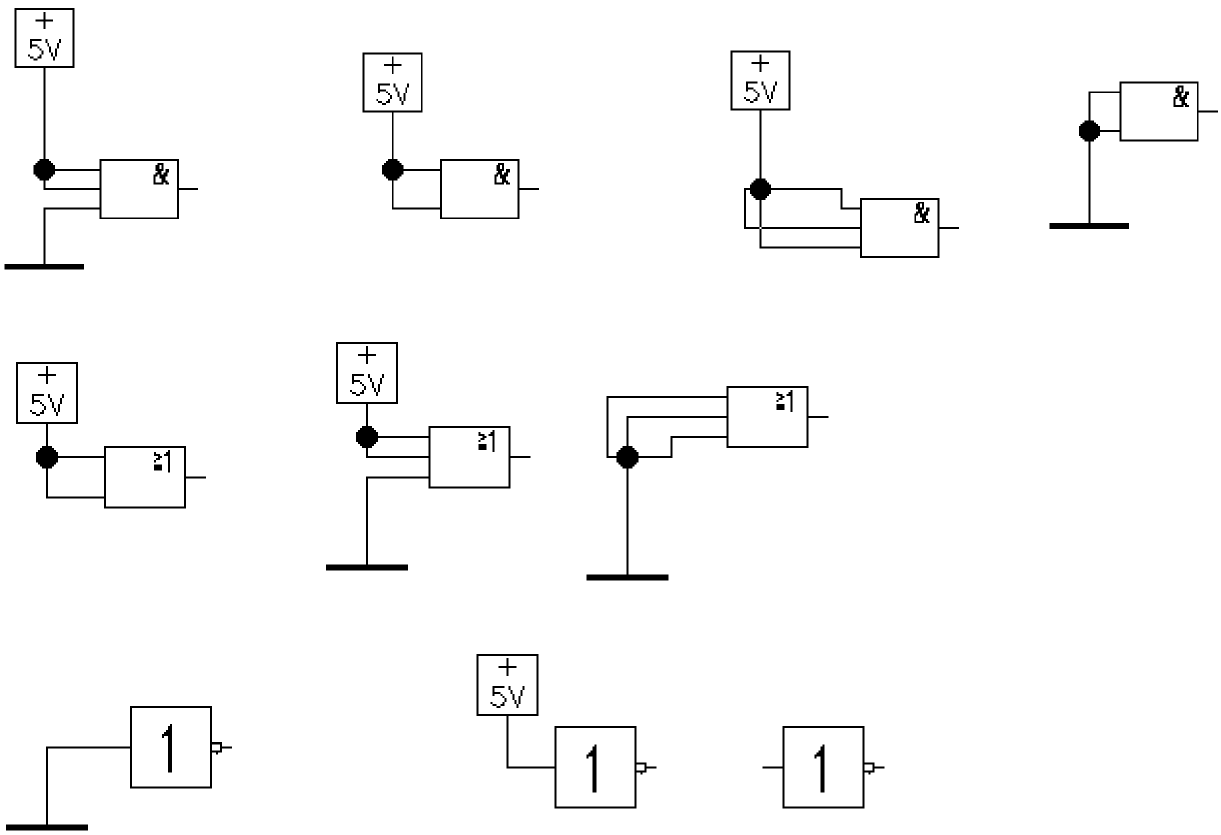
$$a = A + C + B \cdot D + \overline{B} \cdot \overline{D}$$



Preveri svoje znanje:

Preizkusi svoje znanje

1. Kaj dobimo na izhodu naslednjih logičnih vrat?



2. Napiši logično tabelo za logični IN in ALI, če imamo tri vhodne spremenljivke

3. Kako bi s stikalom predstavili funkcijo NE-IN?

4. Podano imamo naslednjo logično funkcijo v obliki tabele:

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

Napiši logično enačbo, jo okrajšaj, nariši funkcijski in krmilni načrt.

5. Sestavi logično tabelo, napiši logično enačbo in nariši vezje za naslednji primer:
Funkcija ima vrednost 1, če imamo vsaj dve izmed treh vhodnih spremenljivk na vrednosti 0.

6. Podana je naslednja logična enačba:

$$Y = AB + C \cdot B + \overline{B} \cdot \overline{A}$$

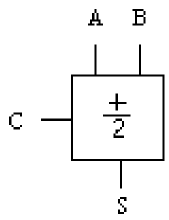
- Napiši logično tabelo
- Okrajšaj logično funkcijo
- Nariši logično vezje

Seštevalnik

Polovični seštevalnik

Polovični seštevalnik zna seštevati le 0+1, 1+0, 0+0 in 1+1. Je logično vezje, kiošteva dve eno-bitni števili

Simbolično ga narišemo kot pravokotnik z dvema vhodoma in dvema izhodoma.



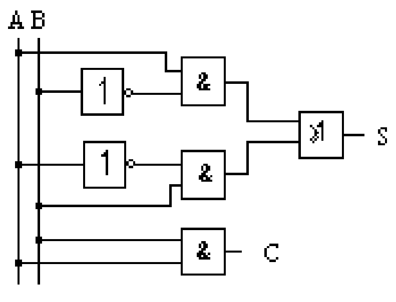
Izjavnostna, pravilnostna ali logična tabela

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

Seštevamo A + B. C pomeni prenos na višje mesto (Carry), S pa suma ali vsota.
Iz tabele napišemo funkciji

$$C = A \,\&\, B \qquad\qquad S = \overline{A}\,B + A\,\overline{B} = A \oplus B$$

in narišemo vezje:



Polni seštevalnik

Sešteva tri eno-bitna števila. Ima tri vhode (A, B in C_i-prenos iz nižjega mesta) in dva izhoda (vsoto S in prenos na višje mesto C_{i+1}).

Tabela:

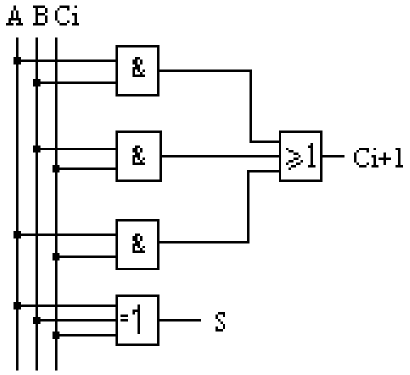
A	B	C_i	C_{i+1}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Imamo dve funkciji treh spremenljivk:

$C_{i+1} = AB + AC_i + BC_i$

$$S = \overline{A}\cdot\overline{B}\cdot C_i + \overline{A}\cdot B\cdot\overline{C_i} + A\cdot\overline{B}\cdot\overline{C}_i + A\cdot B\cdot C_i = A \oplus B \oplus C_i = A \oplus (B \oplus C_i).$$

Narišimo še vezje:



Večbitni seštevalnik

Sešteva dve večbitni števili, npr 3+9. Potrebujemo štiri polne seštevalnike ali tri polne in enega polovičnega. Na vhode pripeljemo števili 3 in 9 v dvojiškem sistemu, na Ci desnega seštevalnika pa 0 (tako dobimo polovični seštevalnik). Na izhodih S dobimo vsoto 3+9=12