Univerza v Ljubljani

Fakulteta za elektrotehniko in računalništvo

PREKLOPNA VEZJA

Priročnik za vaje

Mira TREBAR

Založba FER

PREKLOPNA VEZJA

Priročnik za vaje

Mira TREBAR

LJUBLJANA, 1994

CIP - Kataložni zapis o publikacijah
Narodna in univerzitetna knjižnica, Ljubljana

681.3.02:517.987(075.8)(076.1)

MIRA, Trebar
Preklopna vezja - Priročnik za vaje / Mira, Trebar. - 1. izd.
- Ljubljana: Fakulteta za elektrotehniko in računalništvo, 1994

ISBN 86-7739-072-3

42925568

Recenzent: prof.dr. Andrej Dobnikar
Izdala in založila: Fakulteta za elektrotehniko in računalništvo, 1994
Urednik: mag. Peter Šega
Zunanja oprema: Bizjak Drago, arhitekt
Natisnil: KOPIJA Mavrič, Ljubljana
Naklada: 300 izvodov
1. izdaja

Po mnenju Ministrstva za šolstvo in šport Republike Slovenije št. 415-29/94 z dne 07.09.1994 gre za proizvod, od katerega se plačuje davek od prometa proizvodov po tarifni številki 3.

VSEBINA

1 ŠTEVILA IN BINARNI SISTEM

1.1 Številski sistemi ... 1
1.2 Pretvarjanje števil iz enega sistema v drugega ... 3
1.3 Osnove binarnega računanja ... 6
1.4 Binarne kode ... 8

2 BOOLEOVA ALGEBRA

2.1 Postulati Booleove algebre ... 10
2.2 Izreki Booleove algebre ... 11
2.2.1 Dokazi izrekov s postulati ... 11
2.2.2 Dokazi izrekov z Vennovimi diagrami ... 15
2.2.3 Dokazi izrekov s pravilnostno tabelo ... 15
2.2.4 Dokazi izrekov s kontaktnimi shemami ... 16

3 OSNOVE PREKLOPNIH FUNKCIJ

3.1 Oblike preklopnih funkcij ... 18
3.1.1 Zapis popolnih normalnih oblik preklopnih funkcij ... 19
3.1.2 Zapis preklopnih funkcij v numerični obliki ... 22
3.1.3 Pretvorba preklopnih funkcij iz ene oblike v drugo ... 23
3.1.4 Dualne preklopne funkcije ... 24
3.2. Tehnološke rešitve preklopnih funkcij ... 25
3.2.1 Analiza relejskih kontaktnih shem ... 26
3.2.2 Logične operacije v MOS tehnologiji ... 29
3.2.3 C-MOS družina logičnih vezij ... 34
3.3 Razčlenjevanje preklopnih funkcij (ločenje) ... 35
3.3.1 Razčlenjevanje v smeri DNO oz. PDNO ... 35
3.3.2 Razčlenjevanje v smeri KNO oz. PKNO ... 36
3.4. Elementarne funkcije algebre logike ... 38
3.4.1 Funkcijsko polni sistemi preklopnih funkcij ... 39
3.4.2 Reed-Mullerjeva analitična oblika preklopnih funkcij ... 48
3.4.3 Dvonivojska (normalna) oblika preklopnih funkcij ... 49

4 MINIMIZACIJA PREKLOPNIH FUNKCIJ

4.1 Grafična metoda (Veitchev postopek) ... 53
4.1.1 Zapis MDNO - minimalne disjunktivne normalne oblike ... 54
4.1.2 Zapis MKNO - minimalne konjunktivne normalne oblike ... 55
4.1.3 Zapis MNO - minimalne normalne oblike ... 56
4.1.4 Minimizacija nepopolnih preklopnih funkcij (funkcije z redundancami) ... 58
4.2 Quine-ova metoda minimizacije preklopnih funkcij ... 59

4.3 Določanje glavnih vsebovalnikov iz DNO preklopne funkcije 63
4.3.1 Implikacija in glavni vsebovalniki 63
4.3.2 Iterativna metoda določnja glavnih vsebovalnikov 65
4.4 Izračun vseh neredundatnih disjunktivnih oblik preklopne funkcije (Tison-ova) metoda 67

5 PREKLOPNE FUNKCIJE S POSEBNIMI LASTNOSTMI

5.1 Simetrične preklopne funkcije 70
5.1.1 Ugotavljanje simetričnosti 71
5.1.2 Zapis PDNO za podano simetrično funkcijo 74
5.1.3 Lastnosti simetričnih funkcij 75
5.2 Pragovne preklopne funkcije 77
5.2.1 Linearno ločljive funkcije (pragovne funkcije) 77
5.2.2 Realizacija preklopnih funkcij s pragovnimi operatorji 83
5.3 Verjetnostne preklopne funkcije 85
5.3.1 Določanje verjetnostne preklopne funkcije 85
5.3.2 Povečevanje zanesljivosti preklopnih funkcij 87

6 STRUKTURALNA PREKLOPNA VEZJA

6.1 Strukturalna PDNO in pravilnostna tabela 90
6.2 Operatorji srednje integracije 91
6.2.1 Multipleksor (MX) in demultipleksor (DMX) 91
6.2.2 Kodirnik in dekodirnik 100
6.3. LSI programabilni logični elementi 102

7 SEKVENČNA PREKLOPNA VEZJA

7.1 Pomnilne celice 106
7.1.1 Osnovne strukture pomnilnih celic 107
7.1.2 Sinhronske pomnilne celice 108
7.1.3 Določitev izhodnih funkcij za pomnilne celice 109
7.1.4 Določanje vhodnih funkcij za pomnilne celice (D, T, RS, JK) 110
7.2 Realizacija sekvenčnega vezja 114

8 AVTOMATI

8.1 Ekvivalenca končnih avtomatov 120
8.1.1 Pretvorba Mealy (ME) → Moore (MO) 121
8.1.2 Pretvorba Moore (MO) → Mealy (ME) 122
8.2 Minimizacija avtomatov 123
8.2.1 Minimizacija determinističnih avtomatov 124
8.2.2 Minimizacija nedeterminističnih avtomatov 126
8.3 Realizacija končnega avtomata 129
8.3.1 Postopek za realizacijo končnega avtomata z D pomnilnimi celicami 130
8.3.2 Postopek za realizacijo končnega avtomata z JK pomnilnimi celicami 133

1 ŠTEVILA IN BINARNI SISTEM

V digitalnih sistemih so vsi podatki običajno predstavljeni v binarni (dvo-vrednostni) obliki. Uporabljamo torej podatke oz. števila zapisana v binarnem številskem sistemu. V nadaljevanju si bomo ogledali osnovne značilnosti številskih sistemov in metode pretvarjanja iz enega v drug sistem. Pomemben je tudi zapis podatkov v različnih kodih.

1.1 Številski sistemi

V vsakdanji uporabi je najbolj razširjen desetiški sistem (osnova 10) , ki vsebuje 10 simbolov (0,1,2,3,4,5,6,7,8,9). Število je podano s pozicijskim zapisom tako, da je cifra v sekvenci množena s potenco osnove, ki jo predstavlja ustrezno mesto v številu. Pozicijski zapis desetiškega števila 3428 je

$$3428 = 3 \times 10^3 + 4 \times 10^2 + 2 \times 10^1 + 8 \times 10^0$$

Takšen pozicijski zapis je enostavno razširjen tudi na decimalni del števila, le da so uporabljene negativne potence osnove 10. Pozicijski zapis decimalnega dela števila 0.325 je

$$0.325 = 3 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3}$$

Binarni številski sistem

V binarni predstavitvi informacije je vsak znak predstavljen kot en bit. Sekvenca bitov $b_n, b_{n-1}, \ldots, b_1, b_0$ v binarnem številskem sistemu je celo (integer) število N. Zopet je uporabljen pozicijski zapis s simboloma 0, 1 in potencami osnove 2. Število N poljubne dolžine je predstavljeno kot

$$N = b_n \times 2^n + b_{n-1} \times 2^{n-1} + \ldots + b_1 \times 2^1 + b_0 \times 2^0$$

b_0 - najmanj pomemben bit števila (LSB)

b_n - najbolj pomemben bit števila (MSB)

Sekvenca bitov $b_{-1}, b_{-2}, \ldots, b_{-p}$ v binarnem številskem sistemu je decimalni del števila predstavljen kot .N

$$.N = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \ldots + b_{-p} \times 2^{-p}$$

PRIMER 1.1 Zapis binarnih števil (celo število) N=0110 in N=11101 v desetiškem številskem sistemu.

$$0110 = 0\times 2^3 + 1\times 2^2 + 1\times 2^1 + 0\times 2^0 =$$
$$= 0\times 8 + 1\times 4 + 1\times 2 + 0\times 1 = 6_{10}$$

$$11101 = 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 =$$
$$= 1\times 16 + 1\times 8 + 1\times 4 + 0\times 2 + 1\times 1 = 29_{10}$$

Zapis decimalnega dela binarnih števil .N=.0110 in .N=.101 v desetiškem številskem sistemu.

$$.0110 = 0\times 2^{-1} + 1\times 2^{-2} + 1\times 2^{-3} + 0\times 2^{-4} =$$
$$= 0\times 0.5 + 1\times 0.25 + 1\times 0.125 + 0\times 0.0625 = 0.375_{10}$$

$$.101 = 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} =$$
$$= 1\times 0.5 + 0\times 0.25 + 1\times 0.125 = 0.625_{10}$$

Oktalni številski sistem

Oktalni številski sistem (osnova 8) vsebuje 8 simbolov (0,1,2,3,4,5,6,7). Za zapis takega števila v binarni kodi potrebujemo 3 bite.

Binarna koda	Oktalno število
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

PRIMER 1.2 Zapis oktalnega števila 137 v desetiškem in binarnem številskem sistemu.

$$137_8 = 1\times 8^2 + 3\times 8^1 + 7\times 8^0 = 1\times 64 + 3\times 8 + 7\times 1 = 95_{10}$$
$$137_8 = 001\ 011\ 111 = 1011111_2$$

Zgornjo pretvorbo lahko enostavno preverimo, če binarnemu zapisu števila poiščemo njegovo vrednost v desetiškem zapisu.

$$1011111 = 1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0 =$$
$$= 64 + 16 + 8 + 4 + 2 + 1 = 95_{10}$$

Heksadecimalni številski sistem

Heksadecimalni številski sistem (osnova 15) vsebuje 16 različnih simbolov, kjer so številkam dodane še črke (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F). Zapis števila v heksadecimalnem sistemu vsebuje grupiranje štirih bitov.

Binarna koda	Heksadecimalno število
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

PRIMER 1.3 Zapis heksadecimalnega števila \$3A9 v binarnem in desetiškem številskem sistemu.
Zapis heksadecimalnega števila v binarni kodi je ponavadi grupiran v štiri bite, kot je podano spodaj, lahko pa ga zapišemo tudi negrupirano.

$$\begin{aligned} \$3A9 &= (0011)(1010)(1001) \quad \text{grupirano} \\ &= 001110101001_2 \quad \text{negrupirano} \end{aligned}$$

$$\begin{aligned} 1110101001 &= 1\times 2^9 + 1\times 2^8 + 1\times 2^7 + 1\times 2^5 + 1\times 2^3 + 1\times 2^0 = \\ &= 512 + 256 + 128 + 32 + 8 + 1 = 937_{10} \end{aligned}$$

1.2 Pretvarjanje števil iz enega sistema v drugega

Splošna metoda, ki jo uporabljamo pri pretvarjanju števil iz enega številskega sistema v drugega je naslednja:

1. Celi del (integer) števila in decimalni del števila pretvarjamo ločeno.
2. Ponavljamo postopek deljenja celega dela števila z novo osnovo in uporabimo sekvenco ostankov za določanje novega števila.
3. Ponavljamo množenje decimalnega dela števila z novo osnovo in uporabimo cela števila zmnožka za zapis decimalnega dela novega števila.

PRIMER 1.4 Pretvorite celo število 145 v ekvivalentno binarno število.

Deljenje z 2	Ostanek deljenja	
145		
72	1	
36	0	
18	0	
9	0	
4	1	
2	0	
1	0	↑
0	1	MSB - najbolj pomemben bit

Binarno število N dobimo tako, da zapišemo ostanke deljenja od najbo pomembnega bita - MSB navzgor: N = 10010001.

PRIMER 1.5 Pretvorite decimalno število 14.25 v ekvivalentno binarno število.

Pretvorimo najprej celo število 14 v ekvivalentno binarno število:

Deljenje z 2	Ostanek deljenja	
14		
7	0	
3	1	
1	1	↑
0	1	MSB - najbolj pomemben bit

Celi del binarnega števila N dobimo tako, da zapišemo ostanke deljenja od najb(pomembnega bita - MSB navzgor: N = 1110

Pretvorimo decimalni del števila 0.25 v ekvivalentno binarno obliko. Ponavljar množenje z 2, dokler ni decimalni del zmnožka enak 0, ali do določenega štev binarnih mest, ki ga sami definiramo.

Množenje z 2	Celo število zmnožka	
$0.25 \times 2 = 0.50$	0	MSB - najbolj pomemben bit
$0.50 \times 2 = 1.00$	1	↓
$0.00 \times 2 = 0$		

Postopek je končan, ker je decimalni del števila enak 0. Decimalni del štev zapišemo s celimi števili zmnožka od MSB navzdol: .N = .01

Decimalno število ~~15.25~~ zapišemo kot binarno število: N = 1110.01

14.25

PRIMER 1.6 Pretvorite decimalno število 7.29 v ekvivalentno binarno število z osmimi decimalnimi mesti.

Pretvorimo celo število 7 v ekvivalentno binarno število:

Deljenje z 2	Ostanek deljenja	
7		
3	1	
1	1	↑
0	1	MSB - najbolj pomemben bit

Binarno število N dobimo tako, da zapišemo ostanke deljenja od najbolj pomembnega bita - MSB navzgor: N = 111

Pretvorimo decimalni del števila 0.29 v ekvivalentno binarno obliko. Ponavljamo množenje z 2 dokler ne dobimo 8 decimalnih mest binarnega števila.

Množenje z 2	Celo število zmnožka	
$0.29\times2 = 0.58$	0	MSB - najbolj pomemben bit
$0.58\times2 = 1.16$	1	↓
$0.16\times2 = 0.32$	0	
$0.32\times2 = 0.64$	0	
$0.64\times2 = 1.28$	1	
$0.28\times2 = 0.56$	0	
$0.56\times2 = 1.12$	1	
$0.12\times2 = 0.24$	0	

Postopek računanja decimalnega dela v splošnem ni končan, ker je decimalni del števila pri množenju različen od 0. Ker smo sami določili, da bo ostanek zapisan z osmimi mesti, prenehamo z računanjem. Decimalni del števila zapišemo s celimi števili zmnožka od MSB navzdol: .N = .01001010

Decimalno število 7.29 zapišemo kot binarno število: N = 111.01001010

PRIMER 1.7 Pretvorite decimalno število 19.379 v ekvivalentno oktalno število s štirimi decimalnimi mesti.

Pretvorimo celo število 19 v ekvivalentno oktalno število:

Deljenje z 8	Ostanek deljenja		
19			
2	3		Beremo navzgor: 23_8
0	2	↑	

Pretvorimo decimalni del števila 0.379 v ekvivalentno oktalno obliko. Ponav množenje z 8 dokler ne dobimo 4 decimalna mesta oktalnega števila.

Množenje z 8	Celo število zmnožka		
$0.379 \times 8 = 3.032$	3	↓	
$0.032 \times 8 = 0.256$	0		
$0.256 \times 8 = 2.048$	2		Beremo navzdol: $.3020_8$
$0.048 \times 8 = 0.384$	0		

Postopek računanja decimalnega dela v splošnem ni končan, ker je decimal števila pri množenju različen od 0. Ker smo sami določili, da bo ostanek zapi štirimi mesti, prenehamo z računanjem. Decimalni del števila zapišemo s števili zmnožka navzdol.

Zapis oktalnega števila: $19.379_{10} = 23.3020_8$

Preverimo rezultat tako, da zapišemo decimalno število iz dobljenega okt števila:

Celi del števila: $23_8 = 2 \times 8^1 + 3 \times 8^0 = 2 \times 8 + 3 \times 1 = 19_{10}$

Decimalni del števila:
$$\begin{aligned} .3020 &= 3 \times 8^{-1} + 0 \times 8^{-2} + 2 \times 8^{-3} + 0 \times 8^{-4} = \\ &= 3 \times 0.125 + 2 \times 0.0020 = 0.379_{10} \end{aligned}$$

1.3 Osnove binarnega računanja

Binarna števila lahko predstavimo na dva načina: brez predznaka (unsigned), s predznakom (signed). Števila brez predznaka imajo MSB bit uporablje vrednost. Pri 2-bajtnih številih (16-bitov) imamo v tem primeru zapisana štev 0 do 64436. Števila s predznakom imajo MSB bit določen kot predznak (sign t pove ali so števila pozitivna ali negativna. Pozitivna števila imajo predzı negativna števila pa predznak 1 in so zapisana v 2'K - dvojiškem komplemen 2-bajtnih številih imamo v tem primeru zapisana števila od -32768 do +32767.

Oglejmo si 8-bitna števila brez predznaka in s predznakom.

Število	Brez predznaka	S predznakom
5	00000101	0 0000101
13	00001101	0 0001101
- 5		1 1111010
-13		1 1110011
		↑ - predznak

Komplementarne vrednosti števil

Komplementarne vrednosti števil uporabljamo pri logični operaciji odštevanja , ki je izvedena z isto aparaturno opremo (hardware) kot operacija seštevanja. Operacija odštevanja je izvedena z operacijo seštevanja v komplementarni aritmetiki.

1'K - eniški komplement binarnega števila je število, ki mu zamenjamo ničle z enicami in enice z ničlami

2'K - dvojiški komplement binarnega števila je število, ki ga dobimo tako, da ga pretvorimo v 1'K - eniški komplement in mu prištejemo 1.

PRIMER 1.8
Poiščite 1'K(N) - eniški komplement binarnih števil.

N = 101101	1'K(N) = 010010
N = 001100	1'K(N) = 110011
N = 00101	1'K(N) = 11010

Poiščite 2'K(N) - dvojiški komplement binarnih števil.

N = 101101	2'K(N) = 1'K(N)+1 = 010010 + 1 = 010011
N = 001100	2'K(N) = 1'K(N)+1 = 110011 + 1 = 110100
N = 011101	2'K(N) = 1'K(N)+1 = 100010 + 1 = 100011

Binarno seštevanje

Dve binarni števili, ki ju želimo sešteti morata biti desno poravnani na najmanj pomembnih (LSB) bitih števila. Seštevanje poteka na enak način kot v desetiškem številskem sistemu, le da tu upoštevamo prenos C - "carry" v binarnem številskem sistemu, ki ga dobimo pri seštevanju na i-tem mestu. Pri vsoti 1+1=2 je rezultat seštevanja 0 in prenos C = 1.

PRIMER 1.9 Seštevanje binarnih števil Z = X + Y.

Števili brez predznaka		Števili s predznakom	
X	1001	X	0 0111011
Y	0101	Y	0 0010111
	0001← C		00 111111← C
	1110	Z	0 1010010

Binarno odštevanje

Dve binarni števili, ki sta desno poravnani odštevamo tako, da si sposojamo n višjem mestu. Pri odštevanju na i-tem mestu upoštevamo B - "borrow", ki je enak takrat, ko odštevamo 0-1=1, ker smo si sposodili 2 na višjem mestu števila.

PRIMER 1.10 Odštevanje binarnih števil D = X - Y.

X	1001	X	0101	X	1001
Y	-1001	Y	-0010	Y	-0111
	-000← B		-010← B		-110← B
D	0000	D	0011	D	0010

V praksi je odštevanje izvedeno z isto logiko kot seštevanje, kjer je uporabljen komplementarna aritmetika. Razliko števil X-Y dobimo tako, da zapišemo dvojiš komplement - 2'K(Y), kar pomeni, da smo določili negativno število (-Y) in g prištejemo k številu X. Število je sedaj predstavljeno s predznakom.

PRIMER 1.11 Odštevanje binarnih števil prevedeno v seštevanje števil v 2'K.

Zapišemo število Y v 2'K- dvojiškem komplementu:

Y = 1001	-Y = 2'K(Y) = 1 0111
Y = 0010	-Y = 2'K(Y) = 1 1110
Y = 0111	-Y = 2'K(Y) = 1 1001

Seštejemo število X s številom Y v dvojiškem komplementu - 2'K(Y) = -Y:

X	0 1001	X	0 0101	X	0 1001
-Y	1 0111	-Y	1 1110	-Y	1 1001
	11 111← C		11 100← C		11 001← C
D	0 0000	D	0 0011	D	0 0010

Zadnji prenos seštevanja se v rezultatu ne upošteva.

1.4 Binarne kode

BCD - binarno kodirana desetiška števila so decimalna števila zapisana v ustez binarni obliki. Vsak decimalni simbol je zapisan s 4-bitnim binarnim ekvivalentor Ker imamo samo števila od 0 do 10 v BCD kodi, je 6 binarnih kombinacij pri bitnem zapisu neuporabljenih. Primerjava decimalnega števila in njegovega zapisa BCD.

$$(463)_{10} = (\underbrace{0100}_{4})(\underbrace{0110}_{6})(\underbrace{0011}_{3}) = 010001100011$$

Gray-eva koda je pogosto uporabljena za kodiranje oktalnih ali heksadecimalnih števil, ni pa primerna pri operacijah računanja. Poznamo Gray-evo kodo za poljubno število bitov.

Eno-bitna Gray-eva koda je enaka binarni kodi.

0
1

Dvo-bitna Gray-eva koda je zrcalna slika eno-bitne kode, ki ji je dodana ničla na zgornji polovici zrcalne osi in enica na spodnji polovici.

0 0 Eno-bitna koda
0 1
------
1 1 Zrcalna slika
1 0

Tri-bitna Gray-eva koda je zrcalna slika dvo-bitne kode, ki ji je dodana ničla na zgornji polovici zrcalne osi in enica na spodnji polovici.

0 00
0 01
0 11
0 10
------- Zrcalna os
1 10
1 11
1 01
1 00

Proces generiranja kode ponavljamo za poljubno število bitov.

ASCII koda vključuje poleg kodiranja števil še vse ostale alfanumerične znake, kot so črke in ostali znaki na tastaturi. Omenimo samo, da so decimalni simboli kodirani z 011 pred binarno kodo ustreznega simbola. Celotno tabelo ASCII kode najdemo v številnih priročnikih.

Desetiški simboli	ASCII koda
0	011 0000
1	011 0001
2	011 0010
3	011 0011
4	011 0100
...	....
9	011 1001

2 BOOLEOVA ALGEBRA

Booleova algebra predstavlja osnovo pri študiju preklopnih funkcij in preklopni vezij. Imenujemo jo lahko tudi algebra logike ali preklopna algebra. Zanjo . značilno, da zavzema samo dve vrednosti spremenljivk (0,1) in da so spremenljivk med sabo povezane z operatorji disjunkcije (OR - $\vee$), konjunkcije (AND - &) i negacije (NOT x - $\overline{x}$). Oklepaji so uvedeni za določanje hierarhije operacij.

2.1 Postulati Booleove algebre

V Booleovi algebri postavimo osnovna pravila (postulate) p1, p1', ... , p5, p5' : množico X in operacije (AND, OR, NOT), ki jih uporabljamo pri obravnavan preklopnih funkcij in preklopnih vezij.

- za $x, y \in X$ velja
  - **p1** $x \vee y \in X$
  - **p1'** $x\, y \in X$
- zakon nevtralnih elementov $(0, 1 \in X)$
  - **p2** $x \vee 0 = x$
  - **p2'** $x\, 1 = x$
- zakon komutativnosti za $x, y \in X$
  - **p3** $x \vee y = y \vee x$
  - **p3'** $x\, y = y\, x$
- zakon distributivnosti za $x, y, z \in X$
  - **p4** $x \vee (y\, z) = x \vee y\, z = (x \vee y)\,(x \vee z)$
  - **p4'** $x\,(y \vee z) = (x\, y) \vee (x\, z) = x\, y \vee x\, z$
- zakon antipodnosti za komplementaren element $\overline{x} \in X$
  - **p5** $x \vee \overline{x} = 1$
  - **p5'** $x\, \overline{x} = 0$

Hierarhija logičnih operacij

V vsaki algebri je sekvenca operacij zelo pomembna. Za Booleovo algebro imam določeno naslednjo hierarhijo operacij:

NEGACIJA	prva
AND	↓
OR	zadnja

Vrstni red operacij lahko zamenjamo z uvedbo oklepajev. Najprej so izvede operacije v notranjih oklepajih in se potem izvajajo navzven, vse dokler ni upoš.:vani vsi oklepaji.

2.2 Izreki Booleove algebre

Izreki (teoremi) predstavljajo množico iz postulatov izpeljanih operacij nad binarnimi spremenljivkami, ki jih je vedno možno dokazati s postulati. Poglejmo si najpogosteje uporabljene izreke in njihove dokaze.

Izreki z eno spremenljivko:

$$x \vee 1 = 1$$
$$x\,0 = 0$$
$$x\,x = x$$
$$x \vee x = x$$
$$\overline{\overline{x}} = x$$

Izreki z dvema spremenljivkama:

$$x \vee x\,y = x$$
$$x\,(x \vee y) = x$$
$$(x \vee \overline{y})\,y = x\,y$$
$$x\,\overline{y} \vee y = x \vee y$$
$$(x \vee y) \vee \overline{x} = 1$$
$$\overline{x \vee y} = \overline{x}\,\overline{y} \quad \text{De Morganov izrek}$$
$$\overline{x\,y} = \overline{x} \vee \overline{y} \quad \text{De Morganov izrek}$$

V splošnem lahko De Morganov izrek razširimo na več spremenljivk:

$$\overline{x \vee y \vee \ldots \vee w} = \overline{x}\,\overline{y} \ldots \overline{w}$$
$$\overline{x\,y \ldots w} = \overline{x} \vee \overline{y} \vee \ldots \vee \overline{w}$$

Izreki s tremi spremenljivkami: $x \vee (y \vee z) = (x \vee y) \vee z = x \vee y \vee z$

2.2.1 Dokazi izrekov s postulati

I1: $x \vee 1 = 1$

Dokaz:		
	$x \vee 1 = (x \vee 1)\,1$	p2'
	$= (x \vee 1)\,(x \vee \overline{x})$	p5
	$= x \vee (1\,\overline{x})$	p4
	$= x \vee \overline{x}$	p2', p3'
	$x \vee 1 = 1$	p5

I2: $x\,0 = 0$

Dokaz:		
	$x\,0 = x\,0 \vee 0$	p2
	$= x\,0 \vee x\,\overline{x}$	p5'
	$= x\,(0 \vee \overline{x})$	p4'
	$= x\,\overline{x}$	p2, p3
	$x\,0 = 0$	p5'

I3: $x\,x = x$

$$
\begin{array}{lrll}
\text{Dokaz:} & x\,x & = x\,x \vee 0 & \text{p2} \\
 & & = x\,x \vee x\,\overline{x} & \text{p5'} \\
 & & = x\,(x \vee \overline{x}) & \text{p4'} \\
 & & = x\,1 & \text{p5} \\
 & x\,x & = x & \text{p2'}
\end{array}
$$

I4: $x \vee x = x$

$$
\begin{array}{lrll}
\text{Dokaz:} & x \vee x & = (x \vee x)\,1 & \text{p2'} \\
 & & = (x \vee x)\,(x \vee \overline{x}) & \text{p5} \\
 & & = x \vee x\,\overline{x} & \text{p4} \\
 & & = x \vee 0 & \text{p5'} \\
 & x \vee x & = x & \text{p2}
\end{array}
$$

I5: $\overline{\overline{x}} = x$

Dokaz je možen z uporabo Booleovih konstant 0,1, ki jih zavzame spremenljivka x

$$
x = \begin{cases} 1 \\ 0 \end{cases} \qquad \overline{x} = \begin{cases} \overline{1} = 0 \\ \overline{0} = 1 \end{cases} \qquad \overline{\overline{x}} = \begin{cases} \overline{\overline{1}} = \overline{0} = 1 \\ \overline{\overline{0}} = \overline{1} = 0 \end{cases}
$$

I6: $x \vee x\,y = x$

$$
\begin{array}{lrll}
\text{Dokaz:} & x \vee x\,y & = x\,1 \vee x\,y & \text{p2'} \\
 & & = x\,(1 \vee y) & \text{p4'} \\
 & & = x\,1 & \text{I1} \\
 & x \vee x\,y & = x & \text{p2'}
\end{array}
$$

I7: $x\,(x \vee y) = x$

$$
\begin{array}{lrll}
\text{Dokaz:} & x\,(x \vee y) & = (x \vee 0)\,(x \vee y) & \text{p2} \\
 & & = x \vee 0\,y & \text{p4} \\
 & & = x \vee 0 & \text{I2} \\
 & x\,(x \vee y) & = x & \text{p2}
\end{array}
$$

I8: $(x \vee \overline{y})\,y = x\,y$

$$
\begin{array}{lrll}
\text{Dokaz:} & (x \vee \overline{y})\,y & = y\,x \vee y\,\overline{y} & \text{p4', p3'} \\
 & & = x\,y \vee 0 & \text{p5', p3'} \\
 & (x \vee \overline{y})\,y & = x\,y & \text{p2}
\end{array}
$$

I9: $x\,\overline{y} \vee y = x \vee y$

Dokaz: $y \vee x\,\overline{y} = (y \vee x)\,(y \vee \overline{y})$	p4, p3
$= (x \vee y)\ 1$	p5, p3
$x\,\overline{y} \vee y = x \vee y$	p2'

I10: $(x \vee y) \vee \overline{x} = 1$

Dokaz: $(x \vee y) \vee \overline{x} = 1\ ((x \vee y) \vee \overline{x})$	p2'
$= (x \vee \overline{x})\ ((x \vee y) \vee \overline{x})$	p5
$= (\overline{x} \vee x)\ (\overline{x} \vee (x \vee y))$	p3
$= \overline{x} \vee x\,(x \vee y)$	p4
$= \overline{x} \vee x$	p3, I7
$(x \vee y) \vee \overline{x} = 1$	p5

I11: $(\overline{x}\ \overline{y})\,x = 0$

Dokaz: $(\overline{x}\ \overline{y})\,x = (\overline{x}\ \overline{y})\,x \vee 0$	p2
$= x\,(\overline{x}\ \overline{y}) \vee x\,\overline{x}$	p3', p5'
$= x\,(\overline{x}\ \overline{y} \vee \overline{x})$	p4'
$= x\,\overline{x}$	d1
$(\overline{x}\ \overline{y})\,x = 0$	p5'

d1: $\overline{x} \vee \overline{x}\ \overline{y} = \overline{x}\ 1 \vee \overline{x}\ \overline{y}$	p2'
$= \overline{x}\ (1 \vee \overline{y})$	p4'
$= \overline{x}\ 1$	I1
$= \overline{x}$	p2'

I12: $\overline{x \vee y} = \overline{x}\,\overline{y}$

Predpostavimo, da če naj velja De Morganov izrek za negacijo disjunkcije, potem velja tudi naslednje:

a uvedemo novo spremenljivko z in $\overline{z}$, kjer velja $z = x \vee y$, $\overline{z} = \overline{x}\ \overline{y}$

b po postulatu P5 velja $z \vee \overline{z} = 1$ in po postulatu P5' velja $z\,\overline{z} = 0$

c dokažemo izraza pod (b), če vstavimo za $z = x \vee y$ in za $\overline{z} = \overline{x}\ \overline{y}$, ki pomenita dokaz za De Morganov teorem

$(x \vee y) \vee (\overline{x}\ \overline{y}) = 1$

$(x \vee y)\,(\overline{x}\ \overline{y}) = 0$ **!!! dokazati !!!**

Dokaz: $(x \vee y) \vee (\bar{x}\ \bar{y}) = [(x \vee y) \vee \bar{x}][(x \vee y) \vee \bar{y})]$ p4

$= 1.1$ I10

$= 1$ I3

Dokaz: $(x \vee y)(\bar{x}\ \bar{y}) = (\bar{x}\ \bar{y})(x \vee y)$ p3'

$= (\bar{x}\ \bar{y})\, x \vee (\bar{x}\ \bar{y})\, y$ p4'

$= 0 \vee 0$ I11

$= 0$ I4

I13: $\overline{x\,y} = \bar{x} \vee \bar{y}$

Predpostavimo, da če naj velja De Morganov izrek za negacijo konjunkcije, velja tudi naslednje:

a uvedemo novo spremenljivko z in $\bar{z}$, kjer velja $z = x\,y$, $\bar{z} = \bar{x} \vee \bar{y}$

b po postulatu P5 velja $z \vee \bar{z} = 1$ in po postulatu P5' velja $z\,\bar{z} = 0$

c dokažemo izraza pod (b), če vstavimo za $z = x\,y$ in za $\bar{z} = \bar{x} \vee \bar{y}$, ki po dokaz za De Morganov teorem

$x\,y \vee (\bar{x} \vee \bar{y}) = 1$

$x\,y\,(\bar{x} \vee \bar{y}) = 0$ **!!! dokazati !!!**

I14: $x \vee (y \vee z) = (x \vee y) \vee z = x \vee y \vee z$

Navodilo za dokaz:

a uvedemo novi spremenljivki a in b:

$a = (x \vee y) \vee z$ $b = x \vee (y \vee z)$

$\bar{a} = \overline{(x \vee y) \vee z} = \overline{(x \vee y)}\ \bar{z} = (\bar{x}\ \bar{y})\ \bar{z}$

b izrek 14 velja, če velja $b \vee \bar{a} = 1$

$b \vee \bar{a} = 1$

$= b \vee ((\bar{x}\ \bar{y})\ \bar{z})$ vpis a zgoraj

$= (b \vee (\bar{x}\ \bar{y}))(b \vee \bar{z})$ p4

$= ((b \vee \bar{x})(b \vee \bar{y}))(b \vee \bar{z})$ p4

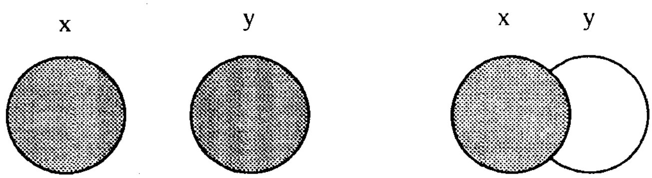
c s postulati in izreki dokažemo, da je vsak izraz znotraj oklepaja ena upoštevamo uvedeno spremenljivko b.

Dokaz izreka nadaljujemo z uporabo postulatov in prej dokazanih izrekov.

2.2.2 Dokazi izrekov z Vennovimi diagrami

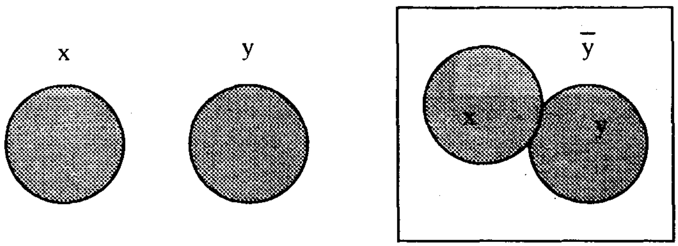
Izrek dokažemo z Vennovimi diagrami tako, da predstavimo spremenljivko s krogom in operacijo disjunkcije z unijo ter operacijo konjunkcije s presekom v množicah.

Dokaz izreka $x \vee x\,y = x$ z Vennovimi diagrami.



$$x \cup (x \cap y) = (x \cup x) \cap (x \cup y) = x$$

Dokaz izreka $x\,\overline{y} \vee y = x \vee y$ z Vennovimi diagrami.



$$(x \cap \overline{y}) \cup y = (x \cup y) \cap (\overline{y} \cup y) = x \cup y$$

2.2.3 Dokazi izrekov s pravilnostno tabelo

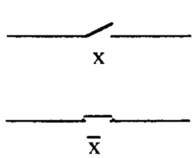
Izrek, ki ga želimo dokazati vpišemo v pravilnostno tabelo. To pomeni, da prevedemo spremenljivke v konstantne vrednosti 0,1 in poiščemo rešitev izreka za vse kombinacije vrednosti, ki jih spremenljivke zavzamejo.

Dokaz izreka $x\,(x \vee y) = x$ s pravilnostno tabelo

x y	x (x ∨ y) = x
0 0	0 0 0
0 1	0 1 0
1 0	1 1 1
1 1	1 1 1

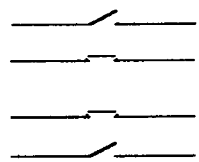
2.2.4 Dokazi izrekov s kontaktnimi shemami

Sklenjen kontakt pri preklopniku ponazarja vrednost spremenljivke 1, odpr vrednost spremenljivke 0.

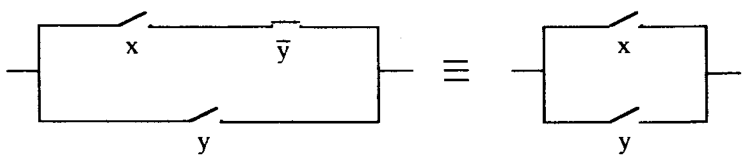


$x = 0$ -odprt
$x = 1$ -zaprt

$x = 0$ $\bar{x} = 1$ -zaprt
$x = 1$ $\bar{x} = 0$ -odprt



Dokaz izreka $x\,\bar{y} \vee y = x \vee y$ s kontaktno shemo



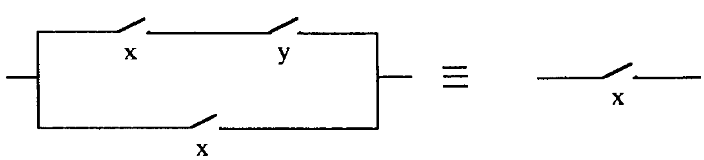
$y = 0 \rightarrow \bar{y} = 1$

stikalo $\bar{y}$ v zgornji veji je zaprto, zato je zgornja veja odvisna od x
spodnja veja je odprta

$y = 1 \rightarrow \bar{y} = 0$

stikalo $\bar{y}$ v zgornji veji je odprto, zato je zgornja veja odprta
spodnja veja je zaprta in rezultat je y

Dokaz izreka $x \vee x\,y = x$ s kontaktno shemo



$x = 0 \rightarrow$ stikali pri x sta v obeh vejah odprti
$x = 1 \rightarrow$ stikalo v spodnji veji je zaprto in rezultat ni odvisen od zgornje veje

3 OSNOVE PREKLOPNIH FUNKCIJ

a

Preklopne funkcije pogosto imenujemo tudi Booleove funkcije ali funkcije algebre logike. Definicija osnovnih izrazov uporabljenih v obravnavi preklopnih funkcij:

- 0,1 - preklopni konstanti
- $x_1, x_2, \ldots, x_n$ - preklopne spremenljivke
- $f(x_1, x_2, \ldots, x_n)$ - preklopna funkcija
- W_i - i-ta vhodna kombinacija ali i-ti vhodni vektor
- $f(W_i)$ - funkcijska vrednost i-te vhodne kombinacije ali i-tega vhodnega vektorja

PREKLOPNA FUNKCIJA $f(x_1, x_2, \ldots, x_n)$ razdeli množico vhodnih kombinacij oz. vhodnih vektorjev v dve podmnožici, kjer je za eno značilna funkcijska vrednost 1 in za drugo funkcijska vrednost 0. Vsaka spremenljivka zavzame v preklopni funkciji vrednost 0 ali 1, zato je preklopna funkcija določena z $2^n - 1$ vhodnimi kombinacijami, ali vhodnimi vektorji W_i in enakim številom funkcijskih vrednosti $f(W_i)$.

PREKLOPNO VEZJE je odločitveno vezje, ki se odloča za eno ali drugo podmnožico preklopnih funkcij. Izhod preklopnega vezja je določen s trenutnimi vrednostmi vhodov, to je vrednostmi preklopnih spremenljivk. Preklopno vezje je sestavljeno iz logičnih operatorjev (vrat).

Preklopno funkcijo lahko predstavimo oz. zapišemo na različne načine. Eden od načinov, ki ga bomo najprej spoznali je zapis preklopne funkcije v pravilnostni tabeli. V levo stran tabele vpišemo vse vhodne kombinacije za n spremenljivk, če le te zavzamejo vrednosti 0, 1. Desna stran tabele pa vsebuje zapis preklopne funkcije s funkcijskimi vrednostmi posamezne vhodne kombinacije.

$x_1\ x_2 \ldots\ x_n$	$f(x_1, x_2, \ldots, x_n)$
W_0	$f(W_0)$
W_1	$f(W_1)$
.	.
W_i	$f(W_i)$
.	.
W_{2^n-2}	$f(W_{2^n-2})$
W_{2^n-1}	$f(W_{2^n-1})$

Drug način predstavitve preklopne funkcije je logična shema, kjer so operatorji posameznega nivoja preklopne funkcije predstavljeni z logičnimi simboli.

3.1 Oblike preklopnih funkcij

Preklopne funkcije so lahko zapisane v različnih oblikah, od katerih so najbolj pogoste naslednje:

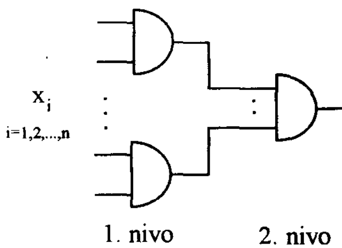
- normalna oblika,
- popolna normalna oblika
- minimalna normalna oblika
- nenormalna oblika

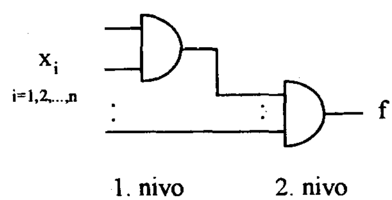
NORMALNA OBLIKA

Preklopna funkcija je zapisana v normalni obliki takrat, kadar obstoja med vhodom, to je spremenljivkami $x_1,x_2,...,x_n$ in izhodom, to je funkcijo $f(x_1,x_2,...,x_n)$ samo en ali največ dva nivoja logičnih operatorjev. Operatorja prvega in drugega nivoja preklopne funkcije sta disjunkcija (v) ali konjunkcija (&).
Oblike normalne preklopne funkcije:

DNO - disjunktivna normalna oblika (1.nivo: &, 2.nivo: v)
KNO - konjunktivna normalna oblika (1.nivo: v, 2.nivo: &)

Logična shema normalne (dvonivojske) oblike preklopnih funcij



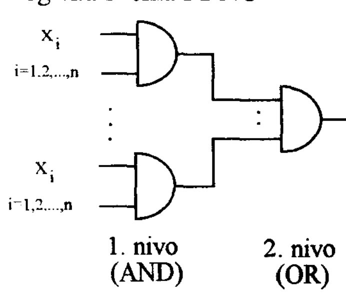


POPOLNA NORMALNA OBLIKA

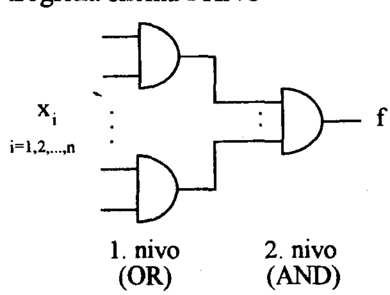
Zapis preklopnih funkcij v popolni normalni obliki pomeni, da ima funkcija dva nivoja operatorjev in vse spremenljivke vstopajo v prvi nivo operatorjev.
Oblike popolne normalne preklopne funkcije:

PDNO - popolna disjunktivna normalna oblika (1.nivo: &, 2.nivo: v)
PKNO - popolna konjunktivna normalna oblika (1.nivo: v, 2.nivo: &)

Logična shema PDNO



Logična shema PKNO



MINIMALNA OBLIKA PREKLOPNE FUNKCIJE

Preklopno funkcijo lahko pogosto poenostavimo (minimiziramo) in obliko, ki jo dobimo imenujemo minimalna oblika.

NENORMALNA OBLIKA PREKLOPNE FUNKCIJE

Kadar obstoja poljubno število nivojev logičnih operatorjev, ki ležijo med neodvisnimi spremenljivkami x_i, kjer je i = 1,2,...,n in odvisno spremenljivko $f(x_1,x_2,...,x_n)$ govorimo o nenormalnih oblikah preklopnih funkcij.

3.1.1 Zapis popolnih normalnih oblik preklopnih funkcij

Spremenljivka x se v preklopnih funkcijah lahko pojavlja kot x in $\bar{x}$. Njen zapis določimo iz vrednosti, ki jo spremenljivka zavzame v posamezni vhodni kombinaciji.

Definicija spremenljivke x :

$$x^w \left\langle \begin{array}{l} \bar{x},\ \text{če je } w = 0 \\ x,\ \text{če je } w = 1 \end{array} \right.$$

Spremenljivka x bo zapisana kot $\bar{x}$, če je w enak konstanti 0 in bo zapisana kot x, če je w enak konstanti 1.

PDNO - Popolna disjunktivna normalna oblika

$$f(x_1,x_2,\ldots,x_n) = \bigvee_{i=0}^{2^n-1} m_i \,\&\, f_i$$

f_i - funkcijska vrednost i-te vhodne kombinacije

m_i - je i-ti minterm, ki je enak konjunktivni povezavi vseh spremenljivk pri i-ti vhodni kombinaciji. Spremenljivka je v mintermu negirana, če ima v pravilnostni tabeli vrednost 0 in nenegirana, če ima v pravilnostni tabeli vrednost 1.

$$m_i = x_1^{w_{i1}} . x_2^{w_{i2}} .... x_n^{w_{in}}$$

PKNO - Popolna konjunktivna normalna oblika

$$f(x_1,x_2,\ldots,x_n) = \mathop{\&}_{i=0}^{2^n-1} (M_{2^n-1-i} \vee f_i)$$

f_i - funkcijska vrednost i-te vhodne kombinacije

M_{2^n-1-i} je 2^n-1-i - ti maksterm, ki je enak disjunkciji vseh spremenljivk 2^n-1-i - te vhodne kombinacije. Spremenljivka je v makstermu negirana, če ima v pravilnostni tabeli vrednost 1 in nenegirana, če ima v pravilnostni tabeli vrednost 0.

$$M_{2^n-1-i} = x_1^{\overline{w}_{i1}} \vee x_2^{\overline{w}_{i2}} \vee \ldots \vee x_n^{\overline{w}_{in}}$$

Med mintermi in makstermi, ki jih zapišemo iz pravilnostne tabele preklopne funkcije veljajo naslednje relacije:

$$\overline{m}_i = M_{2^n-1-i} \qquad \overline{M}_i = m_{2^n-1-i}$$

$$m_i \vee M_{2^n-1-i} = 1 \qquad M_i \vee m_{2^n-1-i} = 1$$

$$m_i \& M_{2^n-1-i} = 0 \qquad M_i \& m_{2^n-1-i} = 0$$

$$m_i \& m_j = 0, \text{ če } i \neq j \qquad M_i \vee M_j = 1, \text{ če } i \neq j$$

PRIMER 3.1 Zapis preklopne funkcije $f(x_1,x_2,x_3)$ v pravilnostni tabeli:

2^n-1-i	i	x_1 x_2 x_3	$f(x_1,x_2,x_3)$
7	0	0 0 0	0
6	1	0 0 1	1
5	2	0 1 0	0
4	3	0 1 1	1
3	4	1 0 0	1
2	5	1 0 1	0
1	6	1 1 0	0
0	7	1 1 1	1

i - indeks mintermov
2^n-1-i - indeks makstermov

Zapis mintermov in makstermov iz pravilnostne tabele:

$$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3 \qquad M_7 = x_1 \vee x_2 \vee x_3$$

$$m_1 = \overline{x}_1\overline{x}_2 x_3 \qquad M_6 = x_1 \vee x_2 \vee \overline{x}_3$$

$$m_2 = \overline{x}_1 x_2\overline{x}_3 \qquad M_5 = x_1 \vee \overline{x}_2 \vee x_3$$

$$m_3 = \overline{x}_1 x_2 x_3 \qquad M_4 = x_1 \vee \overline{x}_2 \vee \overline{x}_3$$

$$m_4 = x_1\overline{x}_2\overline{x}_3 \qquad M_3 = \overline{x}_1 \vee x_2 \vee x_3$$

$$m_5 = x_1\overline{x}_2 x_3 \qquad M_2 = \overline{x}_1 \vee x_2 \vee \overline{x}_3$$

$$m_6 = x_1 x_2\overline{x}_3 \qquad M_1 = \overline{x}_1 \vee \overline{x}_2 \vee x_3$$

$$m_7 = x_1 x_2 x_3 \qquad M_0 = \overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$$

Zapis preklopne funkcije v popolni disjunktivni normalni obliki - PDNO :

$$
\begin{aligned}
f(x_1,x_2,x_3) &= m_0 f_0 \vee m_1 f_1 \vee m_2 f_2 \vee m_3 f_3 \vee m_4 f_4 \vee m_5 f_5 \vee m_6 f_6 \vee m_7 f_7 = \\
&= m_0 0 \vee m_1 1 \vee m_2 0 \vee m_3 1 \vee m_4 1 \vee m_5 0 \vee m_6 0 \vee m_7 1 = \\
&= m_1 1 \vee m_3 1 \vee m_4 1 \vee m_7 1 = \\
&= m_1 \vee m_3 \vee m_4 \vee m_7 = \\
&= \overline{x}_1 \overline{x}_2 x_3 \vee \overline{x}_1 x_2 x_3 \vee x_1 \overline{x}_2 \overline{x}_3 \vee x_1 x_2 x_3
\end{aligned}
$$

Zapis preklopne funkcije v popolni konjunktivni normalni obliki - PKNO :

$$
\begin{aligned}
f(x_1,x_2,x_3) &= (M_7 \vee f_0)(M_6 \vee f_1)(M_5 \vee f_2)(M_4 \vee f_3)(M_3 \vee f_4)(M_2 \vee f_5)(M_1 \vee f_6)(M_0 \vee f_7) = \\
&= (M_7 \vee 0)(M_6 \vee 1)(M_5 \vee 0)(M_4 \vee 1)(M_3 \vee 1)(M_2 \vee 0)(M_1 \vee 0)(M_0 \vee 1) = \\
&= (M_7 \vee 0)(M_5 \vee 0)(M_2 \vee 0)(M_1 \vee 0) = \\
&= M_7 M_5 M_2 M_1 = \\
&= (x_1 \vee x_2 \vee x_3)(x_1 \vee \overline{x}_2 \vee x_3)(\overline{x}_1 \vee x_2 \vee \overline{x}_3)(\overline{x}_1 \vee \overline{x}_2 \vee x_3)
\end{aligned}
$$

Zapis PDNO in PKNO iz normalne oblike z uporabo postulatov

Preklopno funkcijo zapisano v disjunktivni normalni obliki zapišimo v PDNO z uporabo postulatov in izrekov. Vsakemu konjunktivnemu izrazu, ki nima vseh spremenljivk dodamo manjkajoče spremenljivke z uporabo postulatov, kjer uporabimo nevtralen element $x.1 = x(y \vee \overline{y})$, $x \vee 0 = x \vee y\,\overline{y}$ in odpravimo oklepaje.

Preklopno funkcijo v DNO - disjunktivni normalni obliki zapišimo v PDNO z uporabo postulatov.

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

Preklopno funkcijo v DNO - disjunktivni normalni obliki zapišimo v PKNO z uporabo postulatov.

$$
\begin{aligned}
f(x_1,x_2,x_3) &= \overline{x}_1 \vee x_2 x_3 = \\
&= (\overline{x}_1 \vee x_2)(\overline{x}_1 \vee x_3) = \\
&= (\overline{x}_1 \vee x_2 \vee x_3 \overline{x}_3)(\overline{x}_1 \vee x_2 \overline{x}_2 \vee x_3) = \\
&= (\overline{x}_1 \vee x_2 \vee x_3)(\overline{x}_1 \vee x_2 \vee \overline{x}_3)(\overline{x}_1 \vee x_2 \vee x_3)(\overline{x}_1 \vee \overline{x}_2 \vee x_3) = \\
&= (\overline{x}_1 \vee x_2 \vee x_3)(\overline{x}_1 \vee x_2 \vee \overline{x}_3)(\overline{x}_1 \vee \overline{x}_2 \vee x_3)
\end{aligned}
$$

Preklopno funkcijo v KNO - konjunktivni normalni obliki zapišimo v PKNO z uporabo postulatov.

$$
\begin{aligned}
f(x_1,x_2,x_3) &= (x_1 \vee x_2)(\bar{x}_2 \vee x_3) = \\
&= (x_1 \vee x_2 \vee x_3\bar{x}_3)(x_1\bar{x}_1 \vee \bar{x}_2 \vee x_3) = \\
&= (x_1 \vee x_2 \vee x_3)(x_1 \vee x_2 \vee \bar{x}_3)(x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee x_3)
\end{aligned}
$$

NALOGE:
Zapis preklopnih funkcij v PDNO in PKNO:

1. $f(x_1,x_2,x_3) = (x_1 \vee x_2 \vee x_3)$ **R:** $f(x_1,x_2,x_3) = \vee\,(1,2,3,4,5,6,7) = \&\,(7)$
2. $f(x_1,x_2,x_3) = x_1\bar{x}_2 \vee x_2x_3 \vee \bar{x}_1x_3$ **R:** $f(x_1,x_2,x_3) = \vee\,(1,3,4,5,7) = \&\,(7,5,1)$
3. $f(x_1,x_2,x_3) = (x_1 \vee x_3)(x_2 \vee \bar{x}_3)$ **R:** $f(x_1,x_2,x_3) = \vee\,(3,4,6,7) = \&\,(7,6,5,2)$
4. $f(x_1,x_2,x_3) = x_1\bar{x}_2x_3 \vee \bar{x}_3$ **R:** $f(x_1,x_2,x_3) = \vee\,(0,2,4,5,6) = \&\,(6,4,0)$
5. $f(x_1,x_2,x_3) = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee \bar{x}_3)$ **R:** $f(x_1,x_2,x_3) = \vee\,(0,1,4,5,6) = \&\,(5,4,0)$
6. $f(x_1,x_2,x_3) = x_1x_3 \vee \bar{x}_2$ **R:** $f(x_1,x_2,x_3) = \vee\,(0,1,4,5,7) = \&\,(\cancel{7},4,1)$
5

3.1.2 Zapis preklopnih funkcij v numerični obliki

Preklopno funkcijo podano v PDNO ali v PKNO lahko zapišemo krajše v numerični (desetiški) obliki. V oklepaju so za operatorjem disjunkcije ($\vee$) podane desetiške vrednosti mintermov v naraščajočem vrstnem redu pri PDNO in za operatorjem konjunkcije (&) desetiške vrednosti makstermov v padajočem vrstnem redu pri PKNO.

PRIMER 3.2 Zapišite preklopno funkcijo iz pravilnostne tabele v numerični obliki.

2^n-1-i	i	x_1 x_2 x_3	$f(x_1,x_2,x_3)$
7	0	0 0 0	1
6	1	0 0 1	1
5	2	0 1 0	0
4	3	0 1 1	1
3	4	1 0 0	0
2	5	1 0 1	0
1	6	1 1 0	0
0	7	1 1 1	1

PDNO: $f(x_1,x_2,x_3) = \vee\,(0,1,3,7)$
mintermom 0,1,3,7 pripadajo funkcijske vrednosti 1 v tabeli

PKNO: $f(x_1,x_2,x_3) = \&\,(5,3,2,1)$
makstermom 5,3,2,1 pripadajo funkcijske vrednosti 0 v tabeli

z

3.1.3 Pretvorba preklopnih funkcij iz ene oblike v drugo

Če imamo podano preklopno funkcijo v PDNO v numeričnem zapisu, jo lahko pretvorimo v PKNO tako, da jo zapišemo v pravilnostno tabelo in iz nje določimo PKNO. Obstoja pa krajši način pretvorbe brez uporabe pravilnostne tabele.

Pretvorba PDNO → PKNO

V pravilnostni tabeli lahko vidimo, da manjkajoči mintermi v PDNO (pri njih imamo funkcijske vrednosti 0) pretvorjeni v maksterme definirajo PKNO:

manjkajoči mintermi: $m_i \rightarrow M_j; j = 2^n - 1 - i$

PRIMER 3.3 Zapišite PKNO za podano funkcijo v PDNO: $f(x_1,x_2,x_3) = \vee\ (0,3,6)$

Manjkajoči mintermi v zapisu PDNO so tisti mintermi, katerim ustreza funkcijska vrednost 0:

manjkajoči mintermi m_i = 1,2,4,5,7

pretvorba mintermov m_i v maksterme M_j: $j = 2^n-1-i = 7 - i$

mintermi	1	2	4	5	7
makstermi	6	5	3	2	0

Popolno konjunktivno normalno obliko zapišemo z dobljenimi makstermi v padajočem vrstnem redu.

PKNO: $f(x_1,x_2,x_3) = \&\ (6,5,3,2,0)$

Pretvorba PKNO → PDNO

V pravilnostni tabeli lahko vidimo, da manjkajoči makstermi v PKNO (pri njih imamo funkcijske vrednosti 1) pretvorjeni v minterme definirajo PDNO:

manjkajoči makstermi $M_j \rightarrow m_i; i = 2^n - 1 - j$

PRIMER 3.4 Zapišite PDNO za podano funkcijo v PKNO: $f(x_1,x_2,x_3) = \&\ (7,3)$

Manjkajoči makstermi v zapisu PKNO so tisti makstermi, katerim ustreza funkcijska vrednost 1:

manjkajoči makstermi $M_j = 6,5,4,2,1,0$

pretvorba makstermov M_j v minterme $m_i : i = 2^n - 1 - j = 7 - j$

makstermi	6	5	4	2	1	0
mintermi	1	2	3	5	6	7

Popolno disjunktivno normalno obliko zapišemo z dobljenimi mintermi v naraščajočem vrstnem redu.

$$\text{PDNO: } f(x_1,x_2,x_3) = \vee\,(1,2,3,5,6,7)$$

NALOGE:

Pretvorite naslednje preklopne funkcije iz ene popolne oblike v drugo:

1. $f(x_1,x_2,x_3) = \vee\,(0,1,4,6)$ **R:** $f(x_1,x_2,x_3) = \&\,(5,4,2,0)$
2. $f(x_1,x_2,x_3) = \vee\,(1,5,7)$ **R:** $f(x_1,x_2,x_3) = \&\,(7,5,4,3,1)$
3. $f(x_1,x_2,x_3) = \&\,(5,4,0)$ **R:** $f(x_1,x_2,x_3) = \vee\,(0,1,4,5,6)$
4. $f(x_1,x_2,x_3) = \&\,(7,4,3,1,0)$ **R:** $f(x_1,x_2,x_3) = \vee\,(1,2,5)$
5. $f(x_1,x_2,x_3,x_4) = \vee\,(0,1,2,5,8,9)$ **R:** $f(x_1,x_2,x_3,x_4) = \&\,(12,11,9,8,5,4,3,2,1,0)$

3.1.4 Dualne preklopne funkcije

Dualna preklopna funkcija je definirana z negacijo preklopne funkcije $f(x_1,x_2,\ldots,x_n)$ in negacijo nabora spremenljivk $x_1,x_2,\ldots,x_n$.

$$f^d(x_1,x_2,\ldots,x_n) = \bar{f}\,(\bar{x}_1,\bar{x}_2,\ldots,\bar{x}_n)$$

PRIMER 3.5 Zapis dualnih preklopnih funkcij.

$f(x_1,x_2) = x_1 x_2$ $\quad f^d = \overline{\bar{x}_1\bar{x}_2} = x_1 \vee x_2$

Funkcija disjunkcije je dualna funkciji konjunkcije.

$f(x_1,x_2) = x_1 \vee x_2$ $\quad f^d = \overline{\bar{x}_1 \vee \bar{x}_2} = x_1 x_2$

Funkcija konjunkcije je dualna funkciji disjunkcije.

$f(x_1,x_2) = \bar{x}_1\bar{x}_2 \vee x_1 x_2$ $\quad f^d = \overline{\bar{\bar{x}}_1\bar{\bar{x}}_2 \vee \bar{x}_1\bar{x}_2} = (\bar{x}_1 \vee \bar{x}_2)(x_1 \vee x_2) = \bar{x}_1 x_2 \vee x_1\bar{x}_2$

Funkcija EX-OR (seštevanje po mod 2) je dualna funkciji ekvivalence.

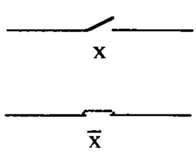
$f(x_1,x_2) = \bar{x}_1 x_2 \vee x_1\bar{x}_2$ $\quad f^d = \overline{\bar{\bar{x}}_1\bar{x}_2 \vee \bar{x}_1\bar{\bar{x}}_2} = (\bar{x}_1 \vee x_2)(x_1 \vee \bar{x}_2) = \bar{x}_1\bar{x}_2 \vee x_1 x_2$

Funkcija ekvivalence je dualna funkciji EX-OR (seštevanje po mod 2).

3.2 Tehnološke rešitve preklopnih funkcij

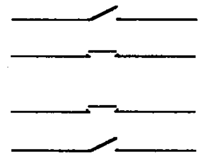
Ogledali si bomo izvedbo AND, OR in NOT operacij z relejskimi kontakti in tranzistorji. Relejski kontakti so najstarejši način za realizacijo logičnih operacij, ki je bil zelo razširjen v industriji. Koncept relejskih vezij je zelo enostaven in primeren za predstavitev logičnih operacij. Uporabimo ga lahko tudi pri analizi tranzistorskih vezij.

Relejski kontakt



$x = 0$ -odprt
$x = 1$ -zaprt

$x = 0$ $\bar{x} = 1$ -zaprt
$x = 1$ $\bar{x} = 0$ -odprt



Paralelna vezava relejskih kontaktov predstavlja logično OR funkcijo in serijska vezava logično AND funkcijo.

Tranzistor

Tranzistorje bomo uporabili kot tipični element pri gradnji elektronskih vrat. Delimo jih v dve skupini:

- bipolarni tranzistorji
- MOSFET tranzistorji, ali krajše MOS.

Če so logična vrata realizirana v tranzistorski tehnologiji, potem imamo opraviti z napetostmi (L - LOW, H - HIGH). Preklopne funkcije, ki jih predstavljamo s spremenljivkami x (x ima vrednosti 0,1), so v različnih tehnologijah predstavljene z dvema nivojema napetosti. Napetost lahko pri obravnavi logičnih funkcij prevedemo v pozitivno logiko - PL ali negativno logiko - NL.

Napetost	PL	NL
LOW	0	1
HIGH	1	0

PRIMER 3.6 Pretvorba funkcije podane z napetostnimi nivoji v pozitivno logiko (PL) in negativno logiko (NL).

x	y	f
L	L	H
L	H	H
H	L	H
H	H	L

PL:

x	y	f=NAND
0	0	1
0	1	1
1	0	1
1	1	0

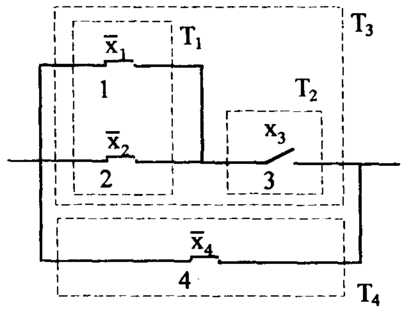
NL:

x	y	f=NOR
1	1	0
1	0	0
0	1	0
0	0	1

3.2.1 Analiza relejskih kontaktnih shem

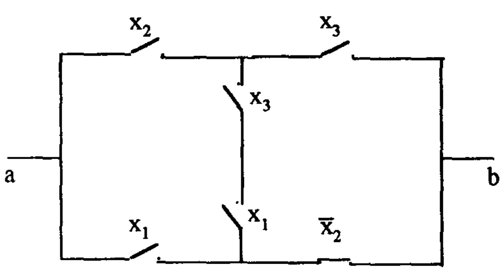
Analiza relejske sheme pomeni opis logičnih lastnosti vezja v obliki pravilnostne tabele, ali preklopne funkcije. Pri zapisu preklopnih funkcij se srečamo s serijsko-paralelnimi vezavami ali vezavami, ki niso serijsko-paralelne.

Zapišimo preklopno funkcijo serijsko - paralelnega vezja.



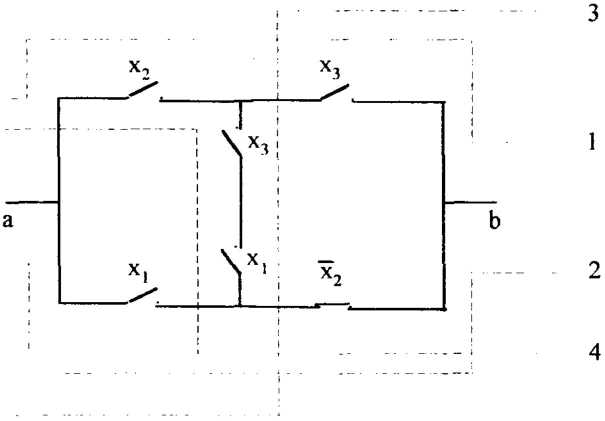
$$f = T_3 \vee T_4 = T_1T_2 \vee T_4 = (\bar{x}_1 \vee \bar{x}_2)x_3 \vee \bar{x}_4$$

Pri vezju, ki ni serijsko-paralelno imamo drugačen pristop zapisa preklopne funkcije. Zapišimo preklopno funkcijo za vezje, ki ni serijsko paralelno.



Iščemo poti, ki povežejo sponki a in b, tako da ne upoštevamo nobene zanke. Vse spremenljivke, ki so na tej poti imenujemo "tie set" tega vezja.

V shemi na naslednji strani imamo vrisane vse možne poti brez zank, ki jih moramo upoštevati pri zapisu funkcije vezja. Primer zapisa poti z oznako 1 za podano shemo je x_2, x_3. Za vsako pot bomo določili spremenljivke, ki jo definirajo ("tie set") in jih zapisali v produkt. Disjunkcija vseh produktov nam določa prenosno funkcijo podanega vezja.



Za vse poti zapišimo "tie sets":

Za pot 1: x_2, x_3 Za pot 3: x_1, x_1, x_3, x_3

Za pot 2: x_1, $\overline{x}_2$ Za pot 4: x_2, x_3, x_1, $\overline{x}_2$

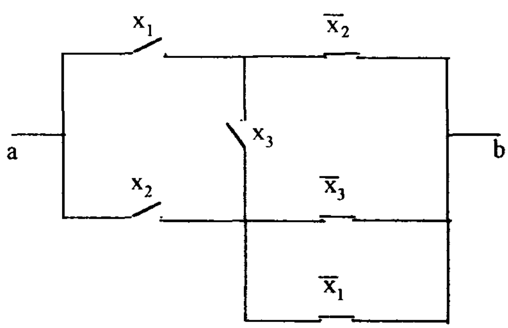
Zapišimo sedaj funkcijo vezja,

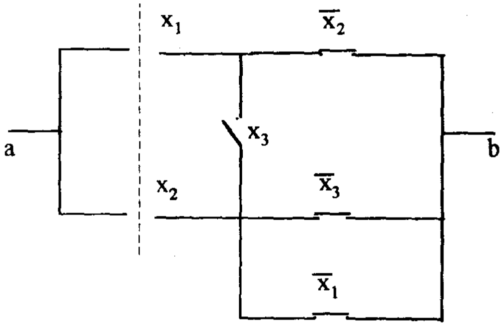
$$f = x_2x_3 \vee x_1\overline{x}_2 \vee x_1x_1x_3x_3 \vee x_2x_3x_1\overline{x}_2$$

Dobljeno funkcijo lahko poenostavimo in rezultat je

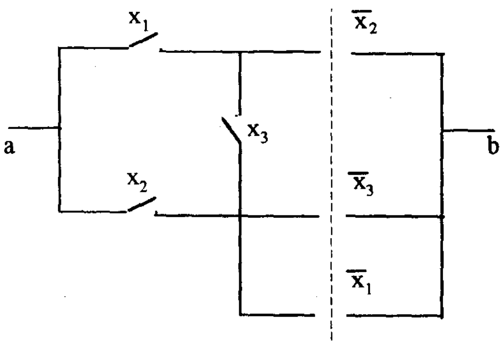
$$f = x_2x_3 \vee x_1\overline{x}_2 \vee x_1x_3$$

Poglejmo si še drug način zapisa vezja, ki ni serijsko-paralelno, kjer poiščemo vse poti, ki razklenejo sponki a in b. Pri tem ne bomo upoštevali zvez med kontakti, kar pomeni da imamo lahko razklenjena kontakta x in $\overline{x}$ istočasno. Celotno vezje je tako razdeljeno v dve izolirani podvezji, kjer ima eno sponko a in drugo sponko b. Če je katerikoli od kontaktov sklenjen bomo dobili obe podvezji sklenjeni.
Spremenljivke, ki določajo takšne kontakte imenujemo "cut sets" podanega vezja. Funkcijo vezja dobimo tako, da disjunktivno povežemo spremenljivke v posameznih "cut sets" in potem disjunktivne izraze povežemo v produkt.



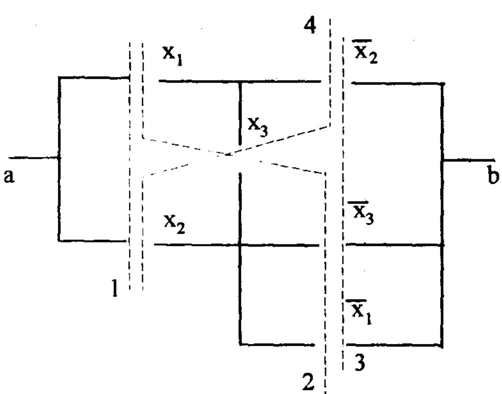


a.



b.

Nadaljujemo postopek iskanja delitve vezja in dobimo naslednje "cut sets":



Zapišimo "cut sets" za odprte kontakte:

Za 1: x_1, x_2 Za 3: $\overline{x}_2$, $\overline{x}_3$, $\overline{x}_1$

Za 2: x_1, x_3, $\overline{x}_3$, $\overline{x}_1$ Za 4: x_2, x_3, $\overline{x}_2$

Zapišimo sedaj funkcijo vezja

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

3.2.2 Logične operacije v MOS tehnologiji

Oglejmo si nekaj osnovnih značilnosti tranzistorjev, ki so osnova elektronskih elementov. Tu se ne bomo spuščali v zgradbo elementov, ampak bomo poskušali pogledati načine povezav posameznih elementov v funkcije. Pri metodah povezovanja tranzistorjev v preklopne funkcije bomo uporabljali

- N-MOS tehnologijo,
- P-MOS tehnologijo.

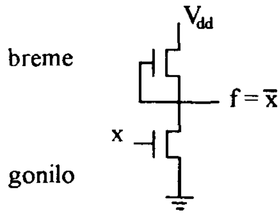
V vsaki tehnologiji lahko definiramo različno logiko

- PL - pozitivna logika,
- NL - negativna logika.

Pri povezovanju tranzistorjev v izbrani tehnologiji in logiki poznamo potem serijsko-paralelno vezavo, kjer je

- S - serijska vezava,
- P - paralelna vezava.

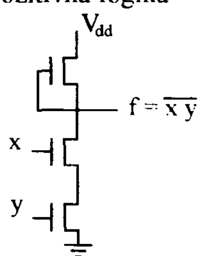
Negator v N-NOS



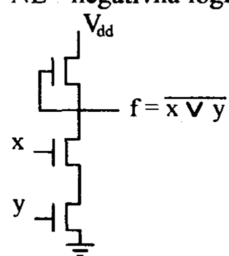
x	$f=\overline{x}$
0	1
1	0

Serijska vezava dveh tranzistorjev v N-MOS

PL - pozitivna logika



NL - negativna logika

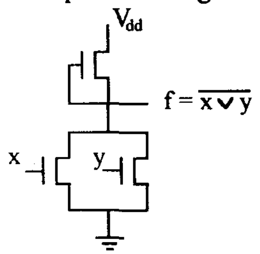


x	y	f = NAND
0	0	1
0	1	1
1	0	1
1	1	0

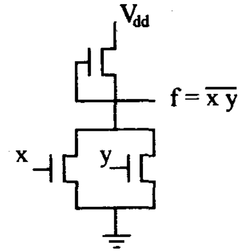
x	y	f = NOR
0	0	1
0	1	0
1	0	0
1	1	0

Paralelna vezava dveh tranzistorjev v N-MOS

PL - pozitivna logika



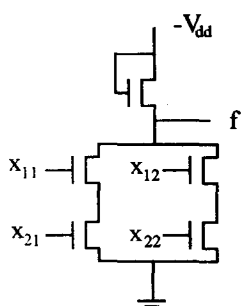
NL - negativna logika

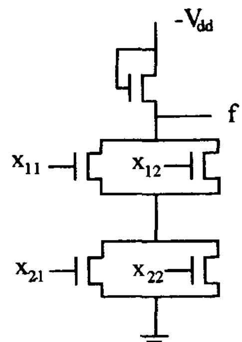


x	y	f = NOR
0	0	1
0	1	0
1	0	0
1	1	0

x	y	f = NAND
0	0	1
0	1	1
1	0	1
1	1	0

Kompleksne MOS celice (P-MOS) - v odvisnosti od uporabljene logike (PL, NL) in načina vezave tranzistorjev (S - serijska, P - paralelna) dobimo različne preklopne funkcije.





Zapis preklopnih funkcij za splošno število tranzistorjev v paralelno/serijski in serijsko/paralelni vezavi. Pomembno je poudariti, da velja negacija izhoda za gonilo kot celoto. Operacije v notranjih vejah tranzistorske vezave preidejo v OR in AND funkcijo.

PL: $f = \overline{\underset{j}{\&}(\underset{i}{\vee} x_{ij})}$

S - OR P - NAND

PL: $f = \overline{\underset{i}{\vee}(\underset{j}{\&} x_{ij})}$

S - NOR P - AND

NL: $f = \overline{\underset{j}{\vee}(\underset{i}{\&} x_{ij})}$

S - NOR P- AND

NL: $f = \overline{\underset{i}{\&}(\underset{j}{\vee} x_{ij})}$

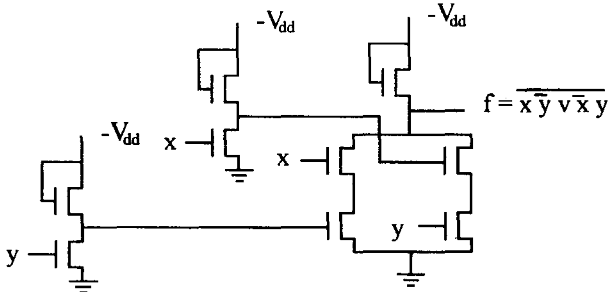
S - NAND P - OR

Določitev operacije za različne kombinacije tehnologije - T (N-MOS , P-MOS), logike (PL, NL) in vezave tranzistorjev -V (S - serijska, P - paralelna)

T	L	V	OPERACIJA	T	L	V
N-MOS	PL	S	NAND	P-MOS	NL	S
N-MOS	PL	P	NOR	P-MOS	NL	P
N-MOS	NL	S	NOR	P-MOS	PL	S
N-MOS	NL	P	NAND	P-MOS	PL	P

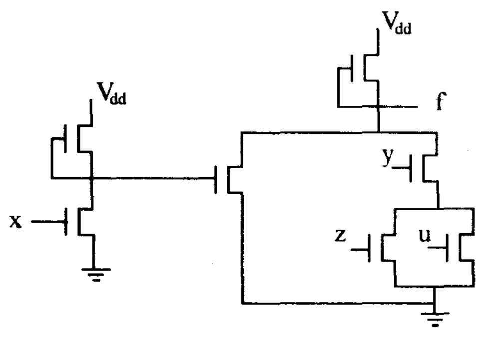
Iz tabele lahko vidimo, da se pri zamenjavi enega od parametrov spremeni logična funkcija in pri zamenjavi dveh parametrov ostane logična funkcija nespremenjena. Če vzamemo za osnovo N-MOS imamo pri PL,S funkcijo NAND in pri PL,P imamo NOR funkcijo, potem dobimo pri zamenjavi tehnologije v P-MOS funkcijo NOR za PL,S in funkcijo NAND za PL,P. Če zamenjamo tehnologijo v P-MOS in logiko v NL dobimo funkcijo NAND za S vezavo in funkcijo NOR za P vezavo, kar je enako kot pri osnovni izbiri.

PRIMER 3.7 Zapišite funkcijo za podano shemo v P - MOS, NL.



Za način vezave tranzistorjev iz tabele določimo, da velja: S - NAND, P - NOR

PRIMER 3.8 Zapišite funkcijo za podano shemo v N - MOS, NL.



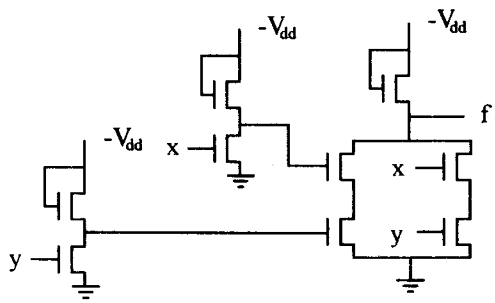
Za način vezave tranzistorjev iz tabele določimo, da velja: S - NOR, P - NAND in dobimo izhodno funkcijo $f = \overline{\overline{x}\,(y \vee z\,u)}$.

PRIMER 3.9 Za podano funkcijo $f = x\,y \vee \bar{x}\,\bar{y}$ definirajte vezje v P - MOS in

Preklopno funkcijo f bomo dvakrat negirali. Eno negacijo v funkciji odpravir De Morganovem pravilu, tako da dobimo negirano preklopno funkcijo, kot jc MOS vezje na izhodu.

$$f = x\,y \vee \bar{x}\,\bar{y} = \overline{\overline{x\,y \vee \bar{x}\,\bar{y}}} = \overline{(\bar{x} \vee \bar{y})(x \vee y)}$$

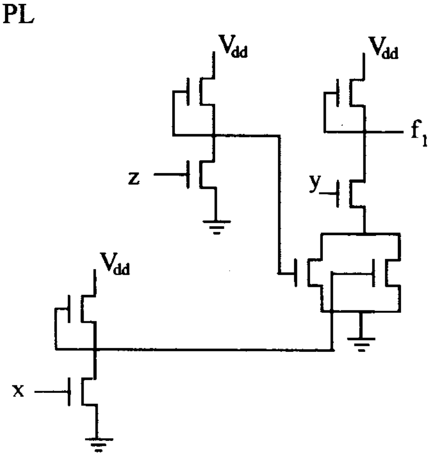
Določitev vezave za P - MOS, PL dobimo iz tabele in velja S - NOR, P - NAND



NALOGE:

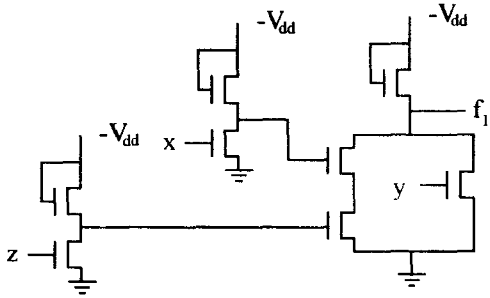
1. Za funkcijo $f_1 = x\,z \leftarrow y = x\,z \vee \bar{y}$ definirajte shemo za:

a. N - MOS, PL



R:

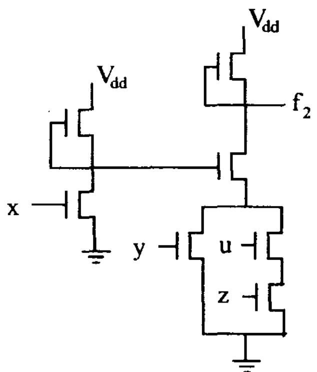
b. P - MOS, PL



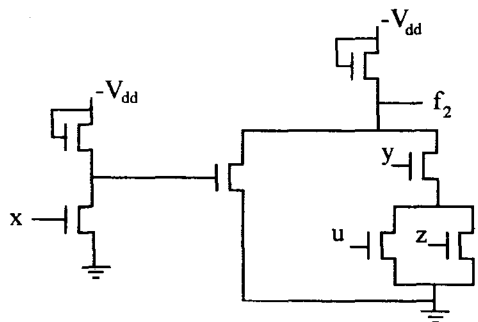
R:

2. Za funkcijo $f_2 = x(\bar{y} \vee \bar{u}\,\bar{z})$ definirajte shemo za:

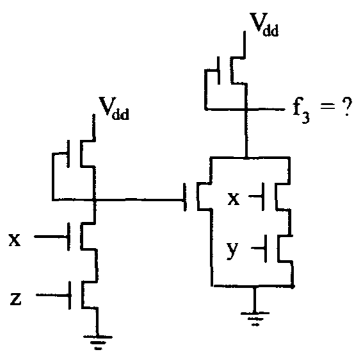
a. N-MOS, NL



b. P-MOS, NL



3. Zapišite DNO za preklopno funkcijo v N - MOS, NL:

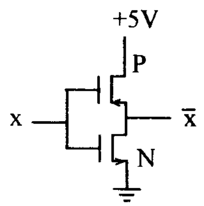


R: $f_3 = x \vee z \vee \bar{x}\,\bar{y}$

3.2.3 C - MOS družina logičnih vezij

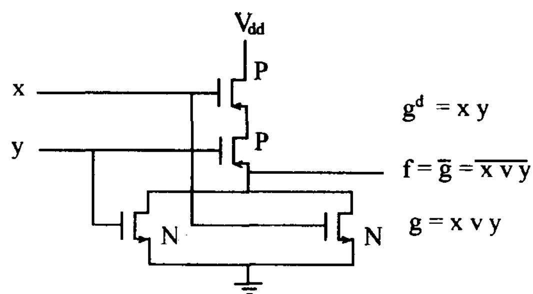
Za C - MOS logična vezja je značilno, da je vsaka vhodna spremenljivka priključena na en P - MOSFET in en N - MOSFET.

Konfiguracija C - MOS inverterja:



C - MOS vrata za splošno funkcijo f se sestoje iz N - MOSFETA in iz P - MOSFETA katerega izhodna funkcija je dualna funkcija k g. Vhodi kontrolirajo tako tranzistorje v gonilu vezja, kakor tudi v bremenu.

Konfiguracija dvovhodnih NOR vrat v C - MOS - u:



Konfiguracija večvhodnih NOR vrat v C-MOS tehnologiji zahteva večje število tranzistorjev v gonilu in na bremenu.

3.3 Razčlenjevanje preklopnih funkcij (ločenje)

Razčlenjevanje preklopnih funkcij je postopek po katerem je funkcija iz normalne oblike prevedena v nenormalno obliko, kjer so dobljeni delni funkcijski ostanki za negirano oz. nenegirano spremenljivko v preklopni funkciji. Funkcijo je možno razčlenjevati po k = 1,...,n . Razčlenitev po k = n generira preklopno funkcijo v popolni normalni obliki.

3.3.1 Razčlenjevanje v smeri DNO oz. PDNO

Funkcija je v razčlenjeni obliki po k spremenljivkah zapisana tako, da disjunktivno povežemo 2^k konjunktivnih členov. Konjunktivni izrazi vsebujejo vse možne zapise spremenljivk po katerih je funkcija razčlenjena in funkcijske ostanke. Funkcijski ostanek dobimo tako, da v funkcijo vstavimo konstantne vrednosti za spremenljivke po katerih funkcijo razčlenjujemo.

$$f(x_1,\dots,x_n) = \bigvee_{i=0}^{2^k-1} x_1^{w_{i1}} \dots x_k^{w_{ik}} . f(w_{1i},\dots,w_{ik},x_{k+1},\dots,x_n)$$

k=1: $f(x_1,\dots,x_i,\dots,x_n) = \overline{x}_i\ f(x_1,\dots,0,\dots,x_n) \vee x_i\ f(x_1,\dots,1,\dots,x_n)$

k=2: $f(x_1,\dots,x_i,x_j,\dots,x_n) = \overline{x}_i\,\overline{x}_j\ f(0,0) \vee \overline{x}_i\,x_j\ f(0,1) \vee x_i\,\overline{x}_j\ f(1,0) \vee x_i\,x_j\ f(1,1)$

Za k = 3,...,n postopek nadaljujemo po opisanem principu.

Funkcijski ostanki:

$f(0,0) = f(x_1,\dots,0,0,\dots,x_n)$ - vstavimo konstantne vrednosti $x_i = 0$, $x_j = 0$

$f(0,1) = f(x_1,\dots,0,1,\dots,x_n)$ - vstavimo konstantne vrednosti $x_i = 0$, $x_j = 1$

$f(1,0) = f(x_1,\dots,1,0,\dots,x_n)$ - vstavimo konstantne vrednosti $x_i = 1$, $x_j = 0$

$f(1,1) = f(x_1,\dots,1,1,\dots,x_n)$ - vstavimo konstantne vrednosti $x_i = 1$, $x_j = 1$

PRIMER 3.10 $f(x_1,x_2,x_3) = x_1x_2 \vee x_1x_3 \vee \overline{x}_1(\overline{x}_2 \vee x_3)$

1. Razčlenitev preklopne funkcije v smeri PDNO po eni spremenljivki: (x_2)

$$\begin{aligned} f(x_1,x_2,x_3) \quad &= \overline{x}_2 f(x_1,0,x_3) \vee x_2 f(x_1,1,x_3) = \\ &= \overline{x}_2(x_1x_3 \vee \overline{x}_1) \vee x_2(x_1 \vee x_1x_3 \vee \overline{x}_1x_3) = \\ &= \overline{x}_2(x_1x_3 \vee \overline{x}_1) \vee x_2(x_1 \vee \overline{x}_1x_3) \end{aligned}$$

2. Razčlenitev preklopne funkcije v smeri PDNO po dveh spremenljivkah: (x_1,x_3)

$$f(x_1,x_2,x_3) \quad = \overline{x}_1\overline{x}_3 f(0,x_2,0) \vee \overline{x}_1x_3 f(0,x_2,1) \vee x_1\overline{x}_3 f(1,x_2,0) \vee x_1x_3 f(1,x_2,1) =$$

$$= \bar{x}_1\bar{x}_3(\bar{x}_2) \vee \bar{x}_1 x_3(1) \vee x_1\bar{x}_3(x_2) \vee x_1 x_3(1)$$
$$= \bar{x}_1\bar{x}_3(\bar{x}_2) \vee \bar{x}_1 x_3 \vee x_1\bar{x}_3(x_2) \vee x_1 x_3$$

3. Razčlenitev preklopne funkcije v smeri PDNO po treh spremenljivkah: (x_1,x_2,x_3)

$$\begin{aligned} f(x_1,x_2,x_3) &= \bar{x}_1\bar{x}_2\bar{x}_3 f(0,0,0) \vee \bar{x}_1\bar{x}_2 x_3 f(0,0,1) \vee \bar{x}_1 x_2\bar{x}_3 f(0,1,0) \vee \bar{x}_1 x_2 x_3 f(0,1,1) \vee \\ &\quad \vee x_1\bar{x}_2\bar{x}_3 f(1,0,0) \vee x_1\bar{x}_2 x_3 f(1,0,1) \vee x_1 x_2\bar{x}_3 f(1,1,0) \vee x_1 x_2 x_3 f(1,1,1) = \\ &= \bar{x}_1\bar{x}_2\bar{x}_3(1) \vee \bar{x}_1\bar{x}_2 x_3(1) \vee \bar{x}_1 x_2\bar{x}_3(0) \vee \bar{x}_1 x_2 x_3(1) \vee \\ &\quad \vee x_1\bar{x}_2\bar{x}_3(0) \vee x_1\bar{x}_2 x_3(1) \vee x_1 x_2\bar{x}_3(1) \vee x_1 x_2 x_3(1) = \\ &= \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2 x_3 \vee \bar{x}_1 x_2 x_3 \vee x_1\bar{x}_2 x_3 \vee x_1 x_2\bar{x}_3 \vee x_1 x_2 x_3 \end{aligned}$$

rezultat je PDNO

3.3.2 Razčlenjevanje v smeri KNO oz. PKNO

Funkcija je v razčlenjeni oblikipo k spremenljivkah zapisana tako, da konjunktivno povežemo 2^k disjunktivnih členov. Disjunktivni izrazi vsebujejo vse možne zapise spremenljivk po katerih je funkcija razčlenjena in funkcijske ostanke. Funkcijski ostanek dobimo tako, da v funkcijo vstavimo konstantne vrednosti za spremenljivke po katerih funkcijo razčlenjujemo.

$$f(x_1,\ldots,x_n) = \mathop{\&}_{i=0}^{2^k-1} (x_1^{w_{i1}} \vee \ldots \vee x_k^{w_{ik}} \vee f(\overline{w}_{i1},\ldots,\overline{w}_{ik},x_{k+1},\ldots,x_n))$$

k=1: $f(x_1,\ldots,x_i,\ldots,x_n) = \{\bar{x}_i \vee f(x_1,\ldots,1,\ldots,x_n)\}\{x_i \vee f(x_1,\ldots,0,\ldots,x_n)\}$

k=2: $f(x_1,\ldots,x_i,x_j,\ldots,x_n) = \{\bar{x}_i \vee \bar{x}_j \vee f(1,1)\}\{\bar{x}_i \vee x_j \vee f(1,0)\}$
$\{x_i \vee \bar{x}_j \vee f(0,1)\}\{x_i \vee x_j \vee f(0,0)\}$

Za k = 3,...,n postopek nadaljujemo po opisanem principu.

Funkcijski ostanki:

$f(0,0) = f(x_1,\ldots,0,0,\ldots,x_n)$ - vstavimo konstantne vrednosti $x_i = 0$, $x_j = 0$

$f(0,1) = f(x_1,\ldots,0,1,\ldots,x_n)$ - vstavimo konstantne vrednosti $x_i = 0$, $x_j = 1$

$f(1,0) = f(x_1,\ldots,1,0,\ldots,x_n)$ - vstavimo konstantne vrednosti $x_i = 1$, $x_j = 0$

$f(1,1) = f(x_1,\ldots,1,1,\ldots,x_n)$ - vstavimo konstantne vrednosti $x_i = 1$, $x_j = 1$

PRIMER 3.11 $f(x_1,x_2,x_3) = x_1 x_2 \vee x_1 x_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1 x_3$

1. Razčlenitev preklopne funkcije v smeri PKNO po eni spremenljivki: (x_1)

$$f(x_1,x_2,x_3) = (\bar{x}_1 \vee f(1,x_2,x_3))\,(x_1 \vee f(0,x_2,x_3)) = (\bar{x}_1 \vee (x_2 \vee x_3))\,(x_1 \vee (\bar{x}_2 \vee x_3))$$

2. Razčlenitev preklopne funkcije v smeri PKNO po dveh spremenljivkah: (x_2,x_3)

$$
\begin{aligned}
f(x_1,x_2,x_3) &= (x_2 \vee x_3 \vee f(x_1,0,0))\,(x_2 \vee \bar{x}_3 \vee f(x_1,0,1)) \\
&\quad (\bar{x}_2 \vee x_3 \vee f(x_1,1,0))\,(\bar{x}_2 \vee \bar{x}_3 \vee f(x_1,1,1)) = \\
&= (x_2 \vee x_3 \vee (\bar{x}_1))\,(x_2 \vee \bar{x}_3 \vee (1))\,(\bar{x}_2 \vee x_3 \vee (x_1))\,(\bar{x}_2 \vee \bar{x}_3 \vee (1)) = \\
&= (x_2 \vee x_3 \vee (\bar{x}_1))\,(\bar{x}_2 \vee x_3 \vee (x_1))
\end{aligned}
$$

3. Razčlenitev preklopne funkcije v smeri PKNO po treh spremenljivkah: (x_1,x_2,x_3)

$$
\begin{aligned}
f(x_1,x_2,x_3) &= (x_1 \vee x_2 \vee x_3 \vee f(0,0,0))\,(x_1 \vee x_2 \vee \bar{x}_3 \vee f(0,0,1))\,(x_1 \vee \bar{x}_2 \vee x_3 \vee f(0,1,0)) \\
&\quad (x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee f(0,1,1))\,(\bar{x}_1 \vee x_2 \vee x_3 \vee f(1,0,0))\,(\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee f(1,0,1)) \\
&\quad (\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee f(1,1,0))\,(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee f(1,1,1)) = \\
&= (x_1 \vee x_2 \vee x_3 \vee 1)\,(x_1 \vee x_2 \vee \bar{x}_3 \vee 1)\,(x_1 \vee \bar{x}_2 \vee x_3 \vee 0)\,(x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee 1) \\
&\quad (\bar{x}_1 \vee x_2 \vee x_3 \vee 0)\,(\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee 1)\,(\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee 1)\,(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee 1) = \\
&= (x_1 \vee \bar{x}_2 \vee x_3)\,(\bar{x}_1 \vee x_2 \vee x_3)
\end{aligned}
$$

rezultat je PKNO

NALOGE:

1. $f(x_1,x_2,x_3) = (x_1 \vee x_2)\,(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$

a. Razčlenite podano preklopno funkcijo v smeri PDNO po spremenljivkah: (x_1,x_3)

R: $f(x_1,x_2,x_3) = \bar{x}_1\bar{x}_3(x_2) \vee \bar{x}_1 x_3(x_2) \vee x_1\bar{x}_3(1) \vee x_1 x_3(\bar{x}_2)$

b. Razčlenite podano preklopno funkcijo v smeri PKNO po spremenljivkah: (x_1,x_3)

R: $f(x_1,x_2,x_3) = (x_1 \vee x_3 \vee (x_2))\,(x_1 \vee \bar{x}_3 \vee (x_2))\,(\bar{x}_1 \vee x_3 \vee (1))\,(\bar{x}_1 \vee \bar{x}_3 \vee (\bar{x}_2))$

2. $f(x_1,x_2,x_3,x_4) = \&\,(\,15,14,12,10,9,6,5,3\,)$

a. Razčleni podano preklopno funkcijo v smeri PDNO po spremenljivkah: (x_2,x_4)

R: $f(x_1,x_2,x_3,x_4) = \bar{x}_2\bar{x}_4(x_1\bar{x}_3 \vee \bar{x}_1 x_3) \vee \bar{x}_2 x_4(x_1 x_3) \vee x_2\bar{x}_4(x_1 x_3 \vee \bar{x}_1\bar{x}_3) \vee x_2 x_4(x_1 \vee x_3)$

b. Razčleni podano preklopno funkcijo v smeri PKNO po spremenljivkah: (x_1,x_4)

R: $f(x_1,x_2,x_3,x_4) = (x_1 \vee x_4 \vee (x_2\bar{x}_3 \vee \bar{x}_2 x_3))\,(x_1 \vee \bar{x}_4 \vee (x_2 x_3))\,(\bar{x}_1 \vee x_4 \vee (x_2 x_3 \vee \bar{x}_2\bar{x}_3))$
$(\bar{x}_1 \vee \bar{x}_4 \vee (x_2 \vee x_3))$

3.4 Elementarne funkcije algebre logike

Za n spremenljivk $x_1, x_2, \ldots, x_n$ je v algebri možno tvoriti 2^{2^n} različnih povezav. Za dve spremenljivki n = 2 imamo 16 različnih funkcij, ki so definirane kot elementarne oz. osnovne funkcije v algebri logike.

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Zapis elementarnih funkcij v normalni obliki:

$f_0 = 0$	preklopna konstanta 0
$f_1 = \bar{x}_1\bar{x}_2 = \overline{x_1 \vee x_2} = x_1 \downarrow x_2$	NOR (Pierce-ova povezava)
$f_2 = \bar{x}_1 x_2 = \overline{x_2 \to x_1}$	negacija implikacije
$f_3 = \bar{x}_1\bar{x}_2 \vee \bar{x}_1 x_2 = \bar{x}_1(\bar{x}_2 \vee x_2) = \bar{x}_1$	negacija spremenljivke x_1
$f_4 = x_1\bar{x}_2 = \overline{x_1 \to x_2}$	negacija implikacije
$f_5 = \bar{x}_1\bar{x}_2 \vee x_1\bar{x}_2 = \bar{x}_2(\bar{x}_1 \vee x_1) = \bar{x}_2$	negacija spremenljivke x_2
$f_6 = \bar{x}_1 x_2 \vee x_1\bar{x}_2 = x_1 \nabla x_2$	EX-OR (seštevanje po mod 2)
$f_7 = \bar{x}_1 \vee \bar{x}_2 = \overline{x_1 x_2} = x_1 \uparrow x_2$	NAND (Sheffer-jeva povezava)
$f_8 = x_1 x_2$	AND (konjunkcija)
$f_9 = \bar{x}_1\bar{x}_2 \vee x_1 x_2 = x_1 \equiv x_2$	EQU - ekvivalenca
$f_{10} = \bar{x}_1 x_2 \vee x_1 x_2 = x_2(\bar{x}_1 \vee x_1) = x_2$	spremenljivka x_2
$f_{11} = \bar{x}_1 \vee x_2 = x_1 \to x_2$	x_1 implicira x_2
$f_{12} = x_1\bar{x}_2 \vee x_1 x_2 = x_1(\bar{x}_2 \vee x_2) = x_1$	spremenljivka x_1
$f_{13} = x_1 \vee \bar{x}_2 = x_2 \to x_1$	x_2 implicira x_1
$f_{14} = x_1 \vee x_2$	OR (disjunkcija)
$f_{15} = 1$	preklopna konstanta 1

Povezave med funkcijami: NAND - NOR, EX-OR - EQU

$\overline{x_1 \uparrow x_2} = x_1 \downarrow x_2$	neg NAND = NOR
$\overline{x_1 \downarrow x_2} = x_1 \uparrow x_2$	neg NOR = NAND
$\overline{x_1 \nabla x_2} = x_1 \equiv x_2$	neg EX-OR = EQU
$\overline{x_1 \equiv x_2} = x_1 \nabla x_2$	neg EQU = EX-OR

3.4.1 Funkcijsko polni sistemi preklopnih funkcij

Funkcijsko poln sistem operatorjev je elementaren sistem operatorjev (&,v,-), ki smo ga spoznali v Booleovi algebri. Z operatorji NAND, NOR, EX-OR, EQU, itd. lahko realiziramo poljubno preklopno funkcijo, če za niz izbranih operatorjev lahko dokažemo, da tvorijo funkcijsko poln sistem.

Zaprti razredi

V logiki obstoja 5 zaprtih razredov, ki jih uporabimo za dokaz funkcijske polnosti nabora operatorjev. Funkcija sodi v zaprti razred takrat, kadar izpolnjuje lastnosti tega razreda. Zaprti razredi so:

T_0 - razred ohranjanja konstante 0
T_1 - razred ohranjanja konstante 1
S - razred sebidualnih funkcij
L - razred linearnih funkcij
M - razred monotonih funkcij

Definicije zaprtih razredov:

T_0 - **Razred ohranjanja konstante 0**

$f \in T_0$, če velja $f(0,0,...,0) = 0$

T_1 - **Razred ohranjanja konstante 1**

$f \in T_1$, če velja $f(1,1,...,1) = 1$

S - Razred sebidualnih preklopnih funkcij
Preklopna funkcija je sebidualna, če je osnovna preklopna funkcija enaka dualni.

$f \in S$, če velja $f(x_1,x_2,...,x_n) = \overline{f}(\overline{x}_1,\overline{x}_2,...,\overline{x}_n)$

PRIMER 3.12 Zapis vseh sebidualnih preklopnih funkcij za spremenljivki x_1, x_2.

x_1	x_2	f_3	f_5	f_{10}	f_{12}
0	0	1	1	0	0
0	1	1	0	1	0
1	0	0	1	0	1
1	1	0	0	1	1

Iz nabora elementarnih preklopnih funkcij so funkcije f_3, f_5, f_{10}, f_{12} z lastnostjo sebidualnosti.

PRIMER 3.13 Ali je preklopna funkcija $f(x_1,x_2) = x_1 x_2 \vee \bar{x}_1 \bar{x}_2$ sebidualna.

Rešitev lahko najdemo na dva načina:

a. analitično

Poiščemo dualno preklopno funkcijo (funkcijo podano v DNO negiramo in negiramo vse njene spremenljivke ter jo poenostavimo). Če je dobljena funkcija enaka prvotni, potem sodi v zaprti razred sebidualnih funkcij.

$$\bar{f}(\bar{x}_1,\bar{x}_2) = \overline{\bar{x}_1\bar{x}_2 \vee x_1x_2} = \overline{\bar{x}_1\bar{x}_2}.\overline{x_1x_2} = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2) = x_1\bar{x}_2 \vee \bar{x}_1x_2$$

$\bar{f}(\bar{x}_1,\bar{x}_2) \neq f(x_1,x_2)$ funkcija ni sebidualna

b. grafično

Pravilnostna tabela je razdeljena v dve polovici, kjer so opazovane spremenljivke od sredine tabele navzven negirane. Če so pri negiranih vrednostih spremenljivk tudi funkcijske vrednosti negirane, potem je preklopna funkcija sebidualna.

$f(x_1,x_2) = x_1 x_2 \vee \bar{x}_1\bar{x}_2$ - funkcija ni sebidualna (enakost funkcijskih vrednosti)

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

$f(x_1,x_2) = \bar{x}_2$ - funkcija je sebidualna, ker ima negirane funkcijske vrednosti v tabeli, če jo opazujemo od sredine navzven pri negiranih vrednostih spremenljivk.

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

L - Razred linearnih preklopnih funkcij

Preklopna funkcija je linearna, če jo je mogoče zapisati kot linearni polinom.

$f \in L$, če velja $\quad f(x_1,x_2,\dots,x_n) = a_0 \nabla a_1x_1 \nabla a_2x_2 \nabla \dots \nabla a_nx_n$

$a_0, a_1, a_2, \dots, a_n$ - koeficienti, ki zavzamejo vrednosti Booleovih konstant 0,1.

PRIMER 3.14 Ali je preklopna funkcija $f(x_1,x_2,x_3) = \&\ (7,3,0)$ linearna.

Rešitev lahko najdemo na dva načina:

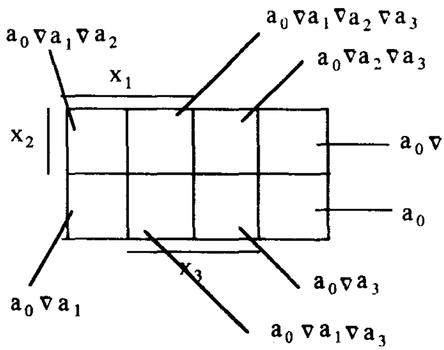
a. analitično - za podano preklopno funkcijo izpišemo 2^n linearnih enačb za izračun koeficientov $a_0,a_1,a_2,\ldots,a_n$. Če so izpolnjene vse enačbe za izračunane koeficiente, potem je preklopna funkcija linerna.

x_1	x_2	x_3	$f(x_1,x_2,x_3)$		
0	0	0	0	$a_0 = 0$	
0	0	1	1	$a_0 \nabla a_3 = 1$	$a_3 = 1$
0	1	0	1	$a_0 \nabla a_2 = 1$	$a_2 = 1$
0	1	1	1	$a_0 \nabla a_2 \nabla a_3 \neq 1$	**ni enako**
1	0	0	0	$a_0 \nabla a_1 = 0$	$a_1 = 0$
1	0	1	1	$a_0 \nabla a_1 \nabla a_3 = 1$	
1	1	0	1	$a_0 \nabla a_1 \nabla a_2 = 1$	
1	1	1	0	$a_0 \nabla a_1 \nabla a_2 \nabla a_3 = 0$	

Enačba $a_0 \nabla a_2 \nabla a_3 = 0 \nabla 1 \nabla 1 \neq 1$ ni izpolnjena za izračunane koeficiente a_0,a_2,a_3, zato preklopna funkcija ni linearna.

b. grafično

Zapis linearnih enačb lahko opazujemo v Veitchevem diagramu. Vsaki funkcijski vrednosti v diagramu ustreza ena od linearnih enačb.



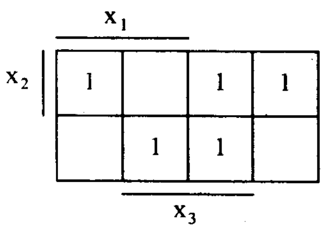
Za ugotavljanje linearnosti je uporabljeno grafično pregibanje likov, kot je prikazano v postopku:

1. Kvadrat s koeficientom a_0 prepognemo proti kvadratu $a_0 \nabla a_n$, v katerem je enaka ali negirana funkcijska vrednost, kar izpolnjuje pogoj za nadaljevanje ugotavljanja linearnosti.

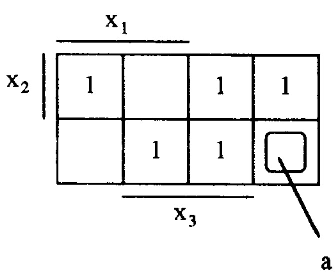
2. Oba zgoraj definirana kvadrata prepognemo proti kvadratoma pri spremenljivki a_{n-1} in če velja enakost ali negacija za združena kvadrata, potem je izpolnjen pogoj za nadaljevanje ugotavljanja linearnosti.
3. Postopek prepogibanja in generiranje večjih likov ponavljamo vse dotlej, dokler ni pokrit cel Veitchev diagram in če je izpolnjen pogoj enakosti ali recipročnosti vse do konca, je preklopna funkcija linearna.

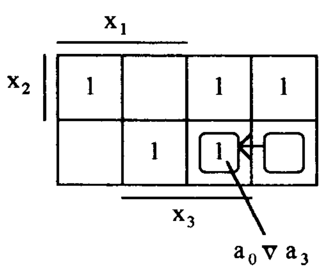
Če je pogoj linearnosti izpolnjen vzamemo n+1 linearnih enačb za izračun koeficientov in jih vstavimo v linearni polinom.

PRIMER 3.15 Za podano preklopno funkcijo v Veitchevem diagramu ugotovite ali je linearna.

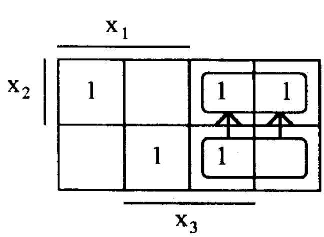


Ugotavljanje linearnosti:



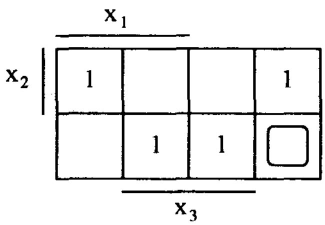


Pri primerjavi prepognjenega kvadrata imamo negirano vrednost, kar pomeni, da je pogoj linearnosti izpolnjen.

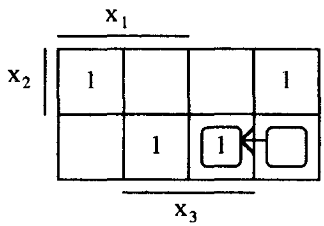


Pogoj enakosti ali recipročnosti (komplementa) ni izpolnjen za označena lika, zato podana preklopna funkcija ni linearna.

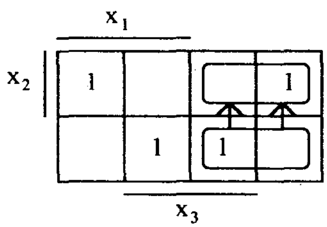
PRIMER 3.16 Ali je preklopna funkcija $f(x_1,x_2,x_3) = \vee\,(1,2,5,6)$ linearna.



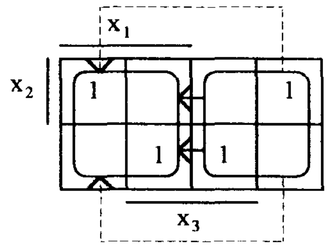
$a_0 = 0$



negirana vrednost:
$a_0 \nabla a_3 = 1$; $a_3 = 1$



negirana vrednost:
$a_0 \nabla a_2 = 1$; $a_2 = 1$



enaka vrednost:
$a_0 \nabla a_1 = 0$; $a_1 = 0$

Pri podani preklopni funkciji je v postopku ugotavljanja linearnosti izpolnjen pogoj enakosti ali recipročnosti za vse možnosti, zato je podana preklopna funkcija linearna. Za izračun koeficintov zadostujejo enačbe, ki so definirane pri posameznem diagramu. Vpišemo izračunane koeficiente v linearni polinom in dobimo zapis preklopne funkcije z linearnim polinomom.

$$f(x_1,x_2,x_3) = a_0 \nabla a_1 x_1 \nabla a_2 x_2 \nabla a_3 x_3 = 0 \nabla 0x_1 \nabla 1x_2 \nabla 1x_3 = x_2 \nabla x_3$$

M - Razred monotonih funkcij

Preklopna funkcija je monotona, če je izpolnjen pogoj primerjave vhodnih kombinacij in funkcijskih vrednosti.

$f \in M$, če velja $\quad w_i \leq w_j \rightarrow f(w_i) \leq f(w_j)$

Pogoj monotonosti mora biti izpolnjen za vse sosedne vhodne vektorje. Vhodna vektorja w_i in w_j sta sosedna, če se razlikujeta samo na enem mestu za 0 in 1, na vseh ostalih pa sta enaka.

w_i	0 0 1 0	0 1 0 1	1 1 0 0
w_j	0 1 1 0	1 0 0 1	0 1 0 0
	$w_i \leq w_j$	ni def.	$w_i \geq w_j$

PRIMER 3.17 Ali je preklopna funkcija $f(x_1,x_2) = x_1 x_2$ monotona.

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

$w_0 \leq w_1, w_2$ $\quad f(w_0) = f(w_1) = f(w_2) = 0$

$w_1 \leq w_3$ $\quad f(w_1) = 0 < f(w_3) = 1$

$w_2 \leq w_3$ $\quad f(w_2) = 0 < f(w_3) = 1$

$f(x_1,x_2)$ je monotona, ker je za vse možnosti primerjave sosednih vhodnih kombinacij izpolnjen pogoj monotonosti.

PRIMER 3.18 Ali je preklopna funkcija v pravinostni tabeli monotona.

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$w_0 \leq w_1, w_2, w_4$ $\quad f(w_0) < f(w_1), f(w_2); f(w_0) = f(w_4)$

$w_1 \leq w_3, w_5$ $\quad f(w_1) = f(w_3) = f(w_5)$

$w_2 \leq w_3, w_6$ $\quad f(w_2) = f(w_3) = f(w_6)$

$w_3 \leq w_7$ $\quad f(w_3) > f(w_7)$!!!

$w_4 \leq w_5, w_6$ $\quad f(w_4) < f(w_5) = f(w_6)$

$w_5 \leq w_7$ $\quad f(w_5) > f(w_7)$!!!

$w_6 \leq w_7$ $\quad f(w_6) > f(w_7)$!!!

Pogoj monotonosti ni izpolnjen za vhodne kombinacije $w_3 \leq w_7, w_5 \leq w_7, w_6 \leq w_7$, zato podana preklopna funkcija ni monotona.

Funkcijsko poln sistem

Elementaren sistem funkcij (v , & , -) je funkcijsko poln sistem, saj je z njim možno zapisati vse preklopne funkcije.

Podan imamo sistem preklopnih funkcij $F = (f_1, f_2, \ldots, f_k)$, za katerega nas zanima ali je funkcijsko poln. Poljuben sistem preklopnih funkcij F je funkcijsko poln:

- če ga je možno opisati z elementarnim sistemom funkcij (v,&,-)
- če vsebuje vsaj eno ali več funkcij $f \in F$, ki ne pripadajo zaprtim razredom, kar pomeni, da z naborom funkcij odpremo vse zaprtosti.

Za $f \in F$ naj velja: $f \notin T_0$

$f \notin T_1$

$f \notin S$

$f \notin L$

$f \notin M$

Tabela pripadnosti zaprtim razredom za vse elementarne funkcije:

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	T_0	∈	/	∈	/	∈	/	∈	/	∈	/	∈	/	∈	/	∈	/
	T_1	/	/	/	/	/	/	/	/	∈	∈	∈	∈	∈	∈	∈	∈
	S	/	/	/	∈	/	∈	/	/	/	/	∈	/	∈	/	/	/
	L	∈	/	/	∈	/	∈	∈	/	/	∈	∈	/	∈	/	/	∈
	M	∈	/	/	/	/	/	/	/	∈	/	∈	/	∈	/	∈	∈

$f_0, f_1, \ldots, f_{15}$ - vse možne elementarne funkcije za 2 spremenljivki x_1, x_2

/ - funkcija ne pripada zaprtemu razredu

∈ - funkcija pripada zaprtemu razredu

PRIMER 3.19 Dokažite, da nabor operatorjev (EX-OR, AND, 1) tvori funkcijsko poln sistem.

Dokaz:

a. s pretvorbo na elementaren sistem operatorjev

EX-OR: $f = x_1 \nabla x_2 = \bar{x}_1 x_2 \vee x_1 \bar{x}_2$

AND: $f = x_1\, x_2$

Funkciji negacije (NEG) in disjunkcije (OR), ki sta poleg konjunkcije v elementarnem sistemu, moramo zapisati z operatorji novega sistema operatorjev.

Negacijo zapišemo s funkcijo EX-OR in uporabo konstante 1.

$$\bar{x}_1 = \bar{x}_1 . 1 \vee x_1 . 0 = x_1 \nabla 1$$

Disjunkcijo zapišemo kot negacijo konjunkcije in uporabimo zgoraj definirano negacijo za zapis.

$$x_1 \vee x_2 = \overline{\bar{x}_1 \bar{x}_2} = ((x_1 \nabla 1)\,(x_2 \nabla 1)) \nabla 1$$

b. z nepripadnostjo zaprtim razredom T_0, T_1, S, L, M

x_1	x_2	EX-OR	AND	1
0	0	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	0	1	1

T_0:

EX-OR $\quad f(0,0) = 0 \quad f \in T_0$

AND $\quad f(0,0) = 0 \quad f \in T_0$

1 $\quad f(0,0) = 1 \quad f \notin T_0$

T_1:

EX-OR $\quad f(1,1) = 0 \quad f \notin T_1$

AND $\quad f(1,1) = 1 \quad f \in T_1$

1 $\quad f(1,1) = 1 \quad f \in T_1$

S:

EX-OR

$$f(x_1,x_2) = \bar{x}_1 x_2 \vee x_1 \bar{x}_2$$

$$\bar{f}(\bar{x}_1,\bar{x}_2) = \overline{\bar{\bar{x}}_1 \bar{x}_2 \vee \bar{x}_1 \bar{\bar{x}}_2} = \overline{x_1 \bar{x}_2}.\overline{\bar{x}_1 x_2} =$$

$$= (\bar{x}_1 \vee x_2)(x_1 \vee \bar{x}_2) =$$

$$= \bar{x}_1 \bar{x}_2 \vee x_1 x_2 \qquad f \notin S$$

AND

$$f(x_1,x_2) = x_1 x_2$$

$$\bar{f}(\bar{x}_1,\bar{x}_2) = \overline{\bar{x}_1 \bar{x}_2} = x_1 \vee x_2 \qquad f \notin S$$

1

$$f(x_1,x_2) = 1$$

$$\bar{f}(\bar{x}_1,\bar{x}_2) = 0 \qquad f \notin S$$

L:

EX-OR

$a_0 = 0$

$a_0 \nabla a_2 = 1 \qquad a_2 = 1$

$a_0 \nabla a_1 = 1 \qquad a_1 = 1$

$a_0 \nabla a_1 \nabla a_2 = 0 \qquad f \in L$

AND

$a_0 = 0$

$a_0 \nabla a_2 = 0 \qquad a_2 = 0$

$a_0 \nabla a_1 = 0 \qquad a_1 = 0$

$a_0 \nabla a_1 \nabla a_2 \neq 1 \qquad f \notin L$

	1	$a_0 = 1$		
		$a_0 \nabla a_2 = 1$	$a_2 = 0$	
		$a_0 \nabla a_1 = 1$	$a_1 = 0$	
		$a_0 \nabla a_1 \nabla a_2 = 1$		$f \in L$

M:	EX-OR	$w_0 \leq w_1, w_2$	$f(w_0) < f(w_1) = f(w_2)$	
		$w_1 \leq w_3$	$f(w_1) > f(w_3)$	$f \notin M$
	AND	$w_0 \leq w_1, w_2$	$f(w_0) = f(w_1) = f(w_2)$	
		$w_1 \leq w_3$	$f(w_1) < f(w_3)$	
		$w_2 \leq w_3$	$f(w_2) < f(w_3)$	$f \in M$
	1	$w_0 \leq w_1, w_2$	$f(w_0) = f(w_1) = f(w_2)$	
		$w_1 \leq w_3$	$f(w_1) = f(w_3)$	
		$w_2 \leq w_3$	$f(w_2) = f(w_3)$	$f \in M$

Zapišemo tabelo pripadnosti zaprtim razredom:

	EX-OR	AND	1
T_0	$\in$	$\in$	$\notin$
T_1	$\notin$	$\in$	$\in$
S	$\notin$	$\notin$	$\notin$
L	$\in$	$\notin$	$\in$
M	$\notin$	$\in$	$\in$

V vsaki vrstici obstoja vsaj en znak nepripadnosti zaprtemu razredu (vsaj en ali več operatorjev ne pripada zaprtim razredom), kar je dokaz, da je zgornji sistem funkcijsko poln.

NALOGE:

1. Dokažite funkcijsko polnost sistemov operatorjev oz. preklopnih funkcij:

1. ($\uparrow$) - NAND - Shefferjev operator
2. ($\downarrow$) - NOR - Pierceov operator
3. ($\rightarrow$, 0)
4. ($x_1 \nabla x_2$, $x_1 x_2 \nabla x_3$)
5. (EQU , v, 0)

3.4.2 Reed - Mullerjeva analitična oblika preklopnih funkcij

Vsako preklopno funkcijo lahko zapišemo v obliki

$$f(x_1, \dots, x_n) = a_0 \nabla a_1 x_1 \nabla \dots \nabla a_n x_n \nabla a_{n+1} x_1 x_2 \nabla \dots \nabla a_{2^n+1} x_1 x_2 \dots x_n$$

a_i - binarne vrednosti koeficientov (konstanti 0,1)

Prevedba poljubne funkcije v Reed-Mullerjevo obliko:

1. Funkcijo zapišemo v popolno disjunktivno normalno obliko - PDNO.
2. Operator disjunkcije ($\vee$) zamenjamo z EX-OR operatorjem (∇).
3. Nadomestimo $\overline{x}_i = x_i \nabla 1$
4. Odpravimo oklepaje

   $x_j(x_i \nabla 1) = x_j x_i \nabla x_j$ $\qquad$ $x_i \nabla x_i = 0$ - dva enaka člena odpadeta

PRIMER 3.20 Zapišite preklopno funkcijo $f(x_1,x_2,x_3) = x_1(x_2 x_3 \vee \overline{x}_3)$ v Reed-Mullerjevi obliki.

$$\begin{aligned} f(x_1,x_2,x_3) &= x_1(x_2 x_3 \vee \overline{x}_3) = x_1 x_2 x_3 \vee x_1 \overline{x}_3 = x_1 x_2 x_3 \vee x_1(x_2 \vee \overline{x}_2)\overline{x}_3 = \\ &= x_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_1 \overline{x}_2 \overline{x}_3 \text{ (PDNO)} \end{aligned}$$

Prevedba funkcije iz PDNO v Reed-Mullerjevo obliko:

$$\begin{aligned} f(x_1,x_2,x_3) &= x_1 x_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_1 \overline{x}_2 \overline{x}_3 \\ &= x_1 x_2 x_3 \nabla x_1 x_2 (x_3 \nabla 1) \nabla x_1 (x_2 \nabla 1)(x_3 \nabla 1) = \\ &= x_1 x_2 x_3 \nabla x_1 x_2 x_3 \nabla x_1 x_2 \nabla x_1 \nabla x_1 x_2 \nabla x_1 x_3 \nabla x_1 x_2 x_3 = \\ &= x_1 x_2 x_3 \nabla x_1 x_3 \nabla x_1 \end{aligned}$$

NALOGE:

Zapišite podane preklopne funkcije v Reed-Mullerjevi obliki:

1. $f(x_1,x_2,x_3) = x_1 x_2 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3$

R: $f(x_1,x_2,x_3) = 1 \nabla x_1 \nabla x_2 \nabla x_3 \nabla x_1 x_3 \nabla x_2 x_3 \nabla x_1 x_2 x_3$

2. $f(x_1,x_2,x_3,x_4) = (x_1 \uparrow \overline{x}_2)(x_2 \equiv x_3)x_4 \vee (\overline{x}_1 \uparrow x_2)\overline{x}_3 x_4$

R: $f(x_1,x_2,x_3,x_4) = x_4 \nabla x_2 x_4 \nabla x_3 x_4 \nabla x_1 x_2 x_4 \nabla x_1 x_2 x_3 x_4$

3. $f(x_1,x_2,x_3) = \overline{x}_2 x_3 \vee x_3 \overline{x}_4 \vee x_2 \overline{x}_3 x_4$

R: $f(x_1,x_2,x_3) = x_3 \nabla x_2 x_4$

3.4.3 Dvonivojska (normalna) oblika preklopnih funkcij

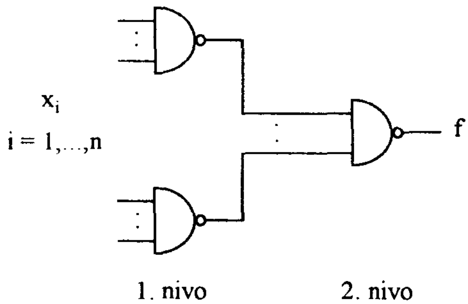
V elementarnem sistemu smo spoznali normalne (dvonivojske) oblike preklopnih funkcij, kjer smo uporabili AND in OR operatorje. Če jima dodamo še NAND in NOR operatorja pridemo do novih dvonivojskih oblik preklopnih funkcij.

PSNO - Popolna Shefferjeva normalna oblika

$$f(x_1,x_2,x_3,x_4) = \mathop{\uparrow}_{i=0}^{2^n-1}(f_i \uparrow s_i)$$

$$s_i = x_1^{w_{i1}} \uparrow x_2^{w_{i2}} \uparrow \ldots \uparrow x_n^{w_{in}} \text{ – Shefferjev minterm}$$

Shefferjev minterm dobimo tako, da mintermu v disjunktivni normalni obliki zamenjamo operator konjunkcije - AND z operatorjem NAND.



PRIMER 3.21 Zapišite podano preklopno funkcijo iz PDNO v PSNO.

PDNO pretvorimo v PSNO tako, da zamenjamo operatorje disjunkcije in konjunkcije na obeh nivojih z NAND operatorji.

PDNO: $f(x_1,x_2,x_3) = x_1\bar{x}_2x_3 \vee x_1x_2x_3$

PSNO: $f(x_1,x_2,x_3) = (x_1 \uparrow \bar{x}_2 \uparrow x_3) \uparrow (x_1 \uparrow x_2 \uparrow x_3)$

PDNO: $f(x_1,x_2,x_3,x_4) = \bar{x}_1x_2x_3x_4 \vee x_1\bar{x}_2x_3\bar{x}_4$

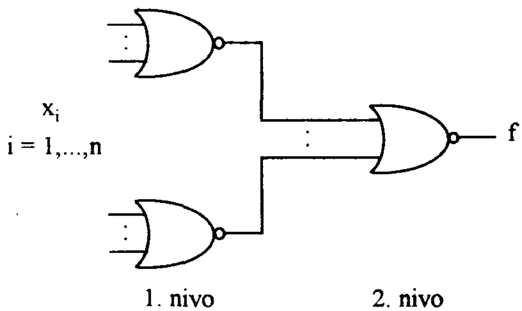
PSNO: $f(x_1,x_2,x_3) = (\bar{x}_1 \uparrow x_2 \uparrow x_3 \uparrow x_4) \uparrow (x_1 \uparrow \bar{x}_2 \uparrow x_3 \uparrow \bar{x}_4)$

PPNO - Popolna Pierceova normalna oblika

$$f(x_1,x_2,\ldots,x_n) = \mathop{\downarrow}_{i=0}^{2^n-1}(f_i \downarrow S_{2^n-1-i})$$

$$s_{2^n-1-i} = x_1^{\bar{w}_{i1}} \downarrow x_2^{\bar{w}_{i2}} \downarrow \ldots \downarrow x_n^{\bar{w}_{in}} \text{ – Pierceov maksterm}$$

Pierceov maksterm dobimo tako, da makstermu pri konjunktivni normalni obliki zamenjamo operator disjunkcije - OR z operatorjem NOR.



PRIMER 3.22 Zapišite podano preklopno funkcijo iz PKNO v PPNO.

PKNO pretvorimo v PPNO tako, da zamenjamo operatorje disjunkcije in konjunkcije na obeh nivojih z NOR operatorji.

PKNO: $f(x_1,x_2,x_3) = (\bar{x}_1 \vee x_2 \vee x_3)(x_1 \vee x_2 \vee \bar{x}_3)$
PPNO: $f(x_1,x_2,x_3) = (\bar{x}_1 \downarrow x_2 \downarrow x_3) \downarrow (x_1 \downarrow x_2 \downarrow \bar{x}_3)$

PKNO: $f(x_1,x_2,x_3,x_4) = \&(12,3) = (x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4)(\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee x_4)$
PPNO: $f(x_1,x_2,x_3,x_4) = (x_1 \downarrow x_2 \downarrow \bar{x}_3 \downarrow \bar{x}_4) \downarrow (\bar{x}_1 \downarrow \bar{x}_2 \downarrow x_3 \downarrow x_4)$

PRIMER 3.23 Zapis DNO in KNO preklopnih funkcij z NAND in NOR operatorji.

DNO: $f(x_1,x_2,x_3) = x_1x_2 \vee \bar{x}_2x_3 \vee \bar{x}_3$
KNO: $f(x_1,x_2,x_3) = (x_1 \vee x_2)(\bar{x}_2 \vee \bar{x}_3)x_3$

Preklopno funkcijo iz DNO ali KNO pretvorimo v funkcijo z NAND ali NOR operatorji tako, da dvakrat negiramo izraz na prvem nivoju, ali pa celotno funkcijo. Negacije rešimo po De Morganovem izreku tako, da negacija AND/OR operatorjev definira NAND ali NOR operacijo.

DNO $\rightarrow$ ($\uparrow$-NAND): $f(x_1,x_2,x_3) = \overline{\overline{x_1x_2 \vee \bar{x}_2x_3 \vee \bar{x}_3}} = \overline{\overline{(x_1x_2)}\;\overline{(\bar{x}_2x_3)}\;\bar{\bar{x}}_3} =$
$= (x_1 \uparrow x_2) \uparrow (\bar{x}_2 \uparrow x_3) \uparrow x_3$

KNO $\rightarrow$ ($\uparrow$-NAND): $f(x_1,x_2,x_3) = \overline{\overline{\overline{\overline{(x_1 \vee x_2)}}\;\overline{\overline{(\bar{x}_2 \vee \bar{x}_3)}}\;x_3}} = \overline{\overline{\overline{\bar{x}_1\bar{x}_2}\;\overline{x_2x_3}\;x_3}} =$
$= \overline{(\bar{x}_1 \uparrow \bar{x}_2) \uparrow (x_2 \uparrow x_3) \uparrow x_3}$

DNO $\rightarrow$ ($\downarrow$-NOR): $f(x_1,x_2,x_3) = \overline{\overline{\overline{\overline{x_1x_2}} \vee \overline{\overline{\bar{x}_2x_3}} \vee \bar{x}_3}} = \overline{\overline{\overline{(\bar{x}_1 \vee \bar{x}_2)} \vee \overline{(x_2 \vee \bar{x}_3)} \vee \bar{x}_3}} =$
$= \overline{(\bar{x}_1 \uparrow \bar{x}_2) \uparrow (x_2 \uparrow \bar{x}_3) \uparrow \bar{x}_3}$

KNO $\rightarrow$ ($\downarrow$-NOR): $f(x_1,x_2,x_3) = \overline{\overline{(x_1 \vee x_2)(\bar{x}_2 \vee \bar{x}_3)x_3}} = \overline{\overline{(x_1 \vee x_2)} \vee \overline{(\bar{x}_2 \vee \bar{x}_3)} \vee \bar{x}_3} =$
$= (x_1 \downarrow x_2) \downarrow (\bar{x}_2 \downarrow \bar{x}_3) \downarrow \bar{x}_3$

Iz nabora operatorjev AND, OR, NAND, NOR je možno določiti 16 različnih oblik preklopnih funkcij, od katerih je samo 8 normalnih (dvonivojskih) oblik, ostale pa imajo še negacijo na izhodu prvega nivoja, ali pa na izhodu drugega nivoja.

Normalne (dvonivojske) oblike so izpeljane iz obeh normalnih oblik (DNO; KNO). V tabeli so prikazane tri dvonivojske oblike (NAND/NAND, NOR/OR, OR/NAND), ki so izpeljane iz DNO in tri dvonivojske oblike (NOR/NOR, NAND/AND, AND/NOR), ki so izpeljane iz KNO.

DNO(AND/OR)	KNO(OR/AND)
NAND/NAND	NOR/NOR
NOR/OR	NAND/AND
OR/NAND	AND/NOR

PRIMER 3.24 Zapis preklopne funkcije v vseh možnih normalnih oblikah.

a. Podana je funkcija v DNO (AND/OR): $f(x_1,x_2,x_3) = x_1x_2 \vee x_1\bar{x}_3 \vee \bar{x}_1x_2x_3$

Normalne oblike zapisane iz DNO so:

NAND/NAND

$$f(x_1,x_2,x_3) = \overline{\overline{x_1x_2 \vee x_1\bar{x}_3 \vee \bar{x}_1x_2x_3}} = \overline{\overline{(x_1x_2)}\;\overline{(x_1\bar{x}_3)}\;\overline{(\bar{x}_1x_2x_3)}} =$$
$$= (x_1 \uparrow x_2) \uparrow (x_1 \uparrow \bar{x}_3) \uparrow (\bar{x}_1 \uparrow x_2 \uparrow x_3)$$

NOR/OR

$$f(x_1,x_2,x_3) = \overline{\overline{x_1x_2}} \vee \overline{\overline{x_1\bar{x}_3}} \vee \overline{\overline{\bar{x}_1x_2x_3}} = \overline{\bar{x}_1 \vee \bar{x}_2} \vee \overline{\bar{x}_1 \vee x_3} \vee \overline{x_1 \vee \bar{x}_2 \vee \bar{x}_3} =$$
$$= (\bar{x}_1 \downarrow \bar{x}_2) \vee (\bar{x}_1 \downarrow x_3) \vee (x_1 \downarrow \bar{x}_2 \downarrow \bar{x}_3)$$

OR/NAND

$$f(x_1,x_2,x_3) = \overline{\overline{x_1x_2 \vee x_1\bar{x}_3 \vee \bar{x}_1x_2x_3}} = \overline{(\bar{x}_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_3)(x_1 \vee \bar{x}_2 \vee \bar{x}_3)} =$$
$$= (x_1 \vee x_2) \uparrow (x_1 \vee \bar{x}_3) \uparrow (\bar{x}_1 \vee x_2 \vee x_3)$$

b. Podana je funkcija v KNO (OR/AND): $f(x_1,x_2,x_3) = (x_1 \vee x_2)(\bar{x}_1 \vee x_2 \vee \bar{x}_3)$

Normalne oblike zapisane iz KNO so:

NOR/NOR

$$f(x_1,x_2,x_3) = \overline{\overline{(x_1 \vee x_2)(\bar{x}_1 \vee x_2 \vee \bar{x}_3)}} = \overline{\overline{(x_1 \vee x_2)} \vee \overline{(\bar{x}_1 \vee x_2 \vee \bar{x}_3)}} =$$
$$= (x_1 \downarrow x_2) \downarrow (\bar{x}_1 \downarrow x_2 \downarrow \bar{x}_3)$$

AND/NOR

$$f(x_1,x_2,x_3) = \overline{\overline{(x_1 \vee x_2)(\bar{x}_1 \vee x_2 \vee \bar{x}_3)}} = \overline{(\bar{x}_1\bar{x}_2) \vee (x_1\bar{x}_2x_3)} =$$
$$= (\bar{x}_1\bar{x}_2) \downarrow (x_1\bar{x}_2x_3)$$

NAND/AND

$$f(x_1,x_2,x_3) = \overline{\overline{(x_1 \vee x_2)}}\;\overline{\overline{(\bar{x}_1 \vee x_2 \vee \bar{x}_3)}} = \overline{(\bar{x}_1\bar{x}_2)}\;\overline{(x_1\bar{x}_2x_3)}$$
$$= (\bar{x}_1 \uparrow \bar{x}_2)(x_1 \uparrow \bar{x}_2 \uparrow x_3)$$

PRIMER 3.25 Zapišite preklopne funkcije v vseh možnih normalnih oblikah.

a. Normalne oblike funkcije iz DNO (AND/OR): $f(x_1,x_2,x_3) = x_1 \vee x_2\bar{x}_3$

NAND/NAND

$$f(x_1,x_2,x_3) = \overline{\overline{x_1 \vee x_2\bar{x}_3}} = \overline{\bar{x}_1\;\overline{(x_2\bar{x}_3)}} = \bar{x}_1 \uparrow (x_2 \uparrow \bar{x}_3)$$

NOR/OR

$$f(x_1,x_2,x_3) = x_1 \vee \overline{\overline{x_2\bar{x}_3}} = x_1 \vee \overline{\bar{x}_2 \vee x_3} = x_1 \vee (\bar{x}_2 \downarrow x_3)$$

OR/NAND

$$f(x_1,x_2,x_3) = \overline{\overline{x_1 \vee x_2\bar{x}_3}} = \overline{\bar{x}_1\overline{(x_2\bar{x}_3)}} = \bar{x}_1 \uparrow (\bar{x}_2 \vee x_3)$$

b. Normalne oblike funkcije iz KNO (OR/AND): $f(x_1,x_2,x_3) = x_2(\bar{x}_1 \vee \bar{x}_2 \vee x_3)$

NOR/NOR

$$f(x_1,x_2,x_3) = \overline{\overline{x_2(\bar{x}_1 \vee \bar{x}_2 \vee_3)}} = \overline{\bar{x}_2 \vee \overline{(\bar{x}_1 \vee \bar{x}_2 \vee x_3)}} = \bar{x}_2 \downarrow (\bar{x}_1 \downarrow \bar{x}_2 \downarrow x_3)$$

AND/NOR

$$f(x_1,x_2,x_3) = \overline{\overline{x_2(\bar{x}_1 \vee \bar{x}_2 \vee_3)}} = \overline{\bar{x}_2 \vee \overline{(\bar{x}_1 \vee \bar{x}_2 \vee x_3)}} = \bar{x}_2 \downarrow (x_1x_2\bar{x}_3)$$

NAND/AND

$$f(x_1,x_2,x_3) = x_2\,\overline{\overline{(\bar{x}_1 \vee \bar{x}_2 \vee x_3)}} = x_2\overline{(x_1x_2\bar{x}_3)} = x_2(x_1 \uparrow x_2 \uparrow \bar{x}_3)$$

4 MINIMIZACIJA PREKLOPNIH FUNKCIJ

Preklopne funkcije v PDNO ali PKNO je v splošnem možno poenostaviti, tako da dobimo krajše oblike preklopnih funkcij. Postopek poenostavljanja preklopnih funkcij imenujemo minimizacija, ki nas pripelje do minimalnih disjunktivnih normalnih oblik - MDNO in do minimalnih konjunktivnih normalnih oblik - MKNO. Konjunktivne izraze, ki jih dobimo v minimizacijskem postopku imenujemo glavni vsebovalniki. Spoznali bomo naslednje metode minimizacije preklopnih funkcij:

Veitchev postopek
Quine-ova metoda
Tison-ova metoda

4.1 Grafična metoda (Veitchev postopek)

Oglejmo si Veitchev diagram za minimizacijo preklopnih funkcij z n spremenljivkami, če je n = 2,3,4. Na mesta mintermov vpisujemo v diagramu pripadajoče funkcijske vrednosti. V spodnjih diagramih so označena mesta mintermov za podano razporeditev spremenljivk.

	x_1	
x_2	m_3	m_1
	m_2	m_0

	x_1	x_1		
x_2	m_6	m_7	m_3	m_2
	m_4	m_5	m_1	m_0
		x_3	x_3	

	x_1	x_1			
x_2	m_{12}	m_{14}	m_6	m_4	
x_2	m_{13}	m_{15}	m_7	m_5	x_4
	m_9	m_{11}	m_3	m_1	x_4
	m_8	m_{10}	m_2	m_0	
		x_3	x_3		

Osnova za poenostavljanje funkcij po grafični metodi je sosednost mintermov oz. konjunktivnih izrazov. Združevanje sosednih konjunkcij vedno izloči tisto spremenljivko, ki se v obeh konjunktivnih izrazih pojavlja kot komplementarna oblika (negirana in nenegirana).

Konjunktivna izraza $x_1 x_2 \overline{x}_3 x_4$ in $x_1 x_2 x_3 x_4$ sta sosedna, ker se spremenljivka x_3 pojavlja v obeh oblikah in ju lahko z uporabo postulatov oz. izrekov Booleove algebre v disjunktivni obliki skrajšamo:

$$x_1 x_2 \overline{x}_3 x_4 \vee x_1 x_2 x_3 x_4 = x_1 x_2 x_4 (\overline{x}_3 \vee x_3) = x_1 x_2 x_4$$

4.1.1 Zapis MDNO - minimalne disjunktivne normalne oblike

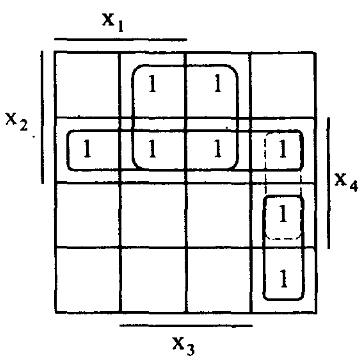
Postopek:

1. Funkcijo vpišemo v Veitchev diagram (vpišemo funkcijske vrednosti 1).

2. Vsak minterm (enico v kvadratu diagrama) primerjamo z vsakim mintermom po sosednosti (razlikovanje i-te spremenljivke za negacijo) in določimo nove konjunktivne izraze ter jih ponovno primerjamo po sosednosti dokler je možno. V primeru ko ne obstoja več sosednosti, je ta končni izraz v diagramu glavni vsebovalnik. To je tak konjunktivni izraz, ki je disjunktivno vsebovan v preklopni funkciji tako, da ne obstaja noben krajši konjunktivni izraz pri opazovanih spremenljivkah x_i, ki bi bil tudi v opazovani funkciji.

3. Za zapis funkcije v MDNO poiščemo samo potrebne glavne vsebovalnike, ki najugodneje pokrijejo minterme v diagramu in jih med sabo disjunktivno povežemo.

PRIMER 4.1 Zapišite funkcijo $f(x_1,x_2,x_3,x_4) = \vee\ (0,1,5,6,7,13,14,15)$ v MDNO.



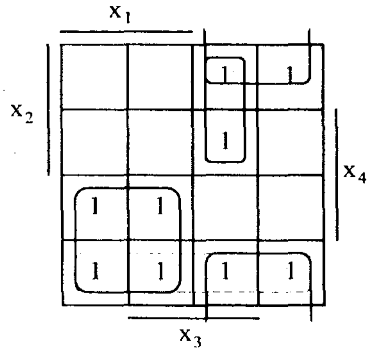
Z združevanjem enic dobimo glavne vsebovalnike: $x_2 x_3$, $x_2 x_4$, $\overline{x}_1 \overline{x}_2 \overline{x}_3$, $\overline{x}_1 \overline{x}_3 x_4$

Za MDNO so potrebni prvi trije glavni vsebovalniki, kajti enici iz glavnega vsebovalnika $\overline{x}_1 \overline{x}_3 x_4$ sta že definirani z ostalimi glavnimi vsebovalniki.

MDNO: $f(x_1,x_2,x_3,x_4) = x_2 x_3 \vee x_2 x_4 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3$

PRIMER 4.2 Zapišite MDNO za funkcijo $f(x_1,x_2,x_3,x_4) = \& \,(14,12,10,3,2,1,0)$.

PDNO: $f(x_1,x_2,x_3,x_4) = \vee\, (0,2,4,6,7,8,9,10,11)$



Z združevanjem enic dobimo glavne vsebovalnike: $x_1\overline{x}_2$, $\overline{x}_2\overline{x}_4$, $\overline{x}_1\overline{x}_4$, $\overline{x}_1 x_2 x_3$

Za MDNO so potrebni samo trije glavni vsebovalniki:

MDNO: $f(x_1,x_2,x_3,x_4) = x_1\overline{x}_2 \vee \overline{x}_1\overline{x}_4 \vee \overline{x}_1 x_2 x_3$

4.1.2 Zapis MKNO -minimalne konjunktivne normalne oblike

Postopek:

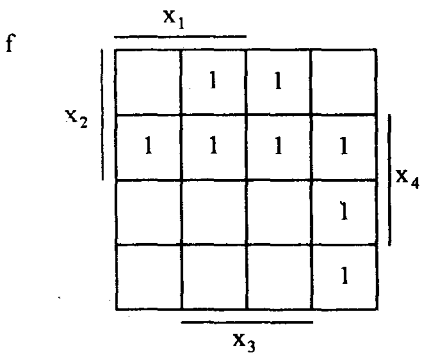
1. Funkcijo v PDNO vpišemo v Veitchev diagram in v drug diagram njeno dopolnilno funkcijo (negacija funkcije f).

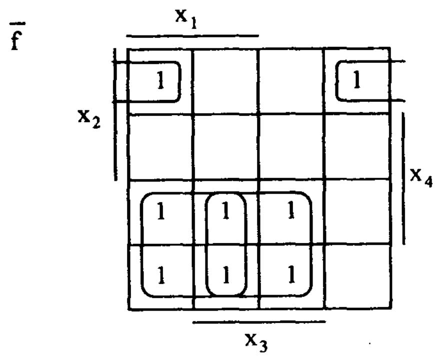
2. Negacijo funkcije poenostavimo v Veitchevem diagramu in dobimo MDNO negirane preklopne funkcije - f (NEG).

3. Poiščemo komplement (negacijo) dopolnilne funkcije v minimalni obliki in rezultat je MKNO.

PRIMER 4.3 Zapišite MKNO za funkcijo $f(x_1,x_2,x_3,x_4) = \vee\, (0,1,5,6,7,13,14,15)$.

1. Zapis funkcije f in $\overline{f}$ v Veitchev diagram:



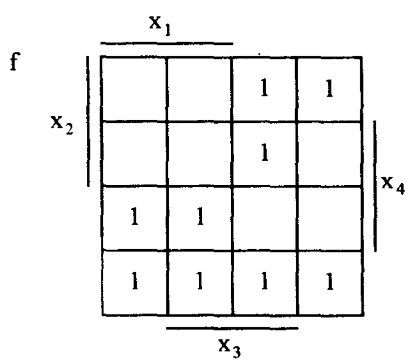


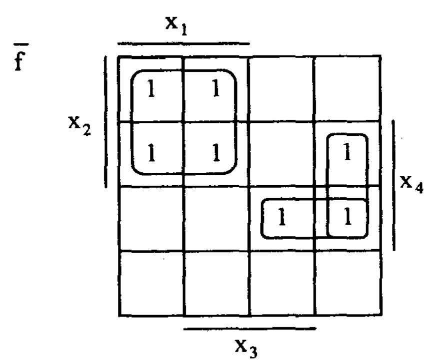
2. MDNO negirane funkcije: f (NEG) = $f(x_1,x_2,x_3,x_4) = x_1\bar{x}_2 \vee \bar{x}_2 x_3 \vee x_2\bar{x}_3\bar{x}_4$

3. MKNO je negacija minimalne oblike funkcije f (NEG):

MKNO: $f(x_1,x_2,x_3,x_4) = \overline{x_1\bar{x}_2 \vee \bar{x}_2 x_3 \vee x_2\bar{x}_3\bar{x}_4} = (\bar{x}_1 \vee x_2)(x_2 \vee \bar{x}_3)(\bar{x}_2 \vee x_3 \vee x_4)$

PRIMER 4.4 Zapišite MKNO za funkcijo $f(x_1,x_2,x_3,x_4) = \vee\,(0,2,4,6,7,8,9,10,11)$.





2. MDNO negirane funkcije: f (NEG) = $f(x_1,x_2,x_3,x_4) = x_1x_2 \vee \bar{x}_1\bar{x}_3x_4 \vee \bar{x}_1\bar{x}_2x_4$

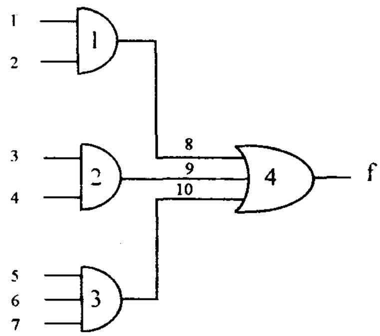
3. MKNO je negacija minimalne oblike funkcije f (NEG):

MKNO: $f(x_1,x_2,x_3,x_4) = \overline{x_1x_2 \vee \bar{x}_1\bar{x}_3x_4 \vee \bar{x}_1\bar{x}_2x_4} = (\bar{x}_1 \vee \bar{x}_2)(x_1 \vee x_3 \vee \bar{x}_4)(x_1 \vee x_2 \vee \bar{x}_4)$

4.1.3 Določitev MNO - minimalne normalne oblike

Minimalna normalna oblika (MNO) je preklopna funkcija zapisana v tisti minimalni obliki (MDNO ali MKNO), ki je krajša po številu operatorjev in/ali številu vhodov. Število operatorjev določimo tako, da preštejemo vse operatorje prvega nivoja in dodamo en operator drugega nivoja. Število vhodov določimo tako, da preštejemo vse vhode v operatorje prvega nivoja in vhode v operator drugega nivoja. Za MNO izberemo tisto obliko, ki ima manj operatorjev. Če je število operatorjev enako, izberemo za MNO tisto obliko, ki ima manj vhodov. V primeru enakega števila operatorjev in vhodov sta obe obliki enakovredni.

PRIMER 4.5 Določite število operatorjev in število vhodov za funkcijo f v logični shemi.



(št.operat. , št. vhodov) = (4,10)

PRIMER 4.6 Poiščite MNO iz MDNO in MKNO za podano funkcijo
$f(x_1,x_2,x_3,x_4) = \vee\,(0,1,5,6,7,13,14,15)$.

MDNO: $f(x_1,x_2,x_3,x_4) = x_2x_3 \vee x_2x_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3$

3 konj op + 1 disj op = 4 operatorji
vhodi: 1.nivo: 7 2.nivo: 3 (4,10)

MKNO: $f(x_1,x_2,x_3,x_4) = (\bar{x}_1 \vee x_2)(x_2 \vee \bar{x}_3)(\bar{x}_2 \vee x_3 \vee x_4)$

3 disj op + 1 konj op = 4 operatorji
vhodi: 1.nivo: 7 2.nivo: 3 (4,10)

MNO = MDNO = MKNO - Obe minimalni obliki preklopne funkcije sta enakovredni.

PRIMER 4.7 Poiščite MNO iz MDNO in MKNO za podano funkcijo
$f(x_1,x_2,x_3,x_4) = \&\,(14,12,10,3,2,1,0)$

MDNO: $f(x_1,x_2,x_3,x_4) = x_1\bar{x}_2 \vee \bar{x}_1\bar{x}_4 \vee \bar{x}_1x_2x_3$

3 konj op + 1 disj op = 4 operatorji
vhodi: 1.nivo: 7 2.nivo: 3 (4,10)

MKNO: $f(x_1,x_2,x_3,x_4) = (\overline{x}_1 \vee \overline{x}_2)(x_1 \vee x_3 \vee \overline{x}_4)(x_1 \vee x_2 \vee \overline{x}_4)$

3 disj op + 1 konj op = 4 operatorji
vhodi: 1.nivo: 8 2.nivo: 3 (4,11)

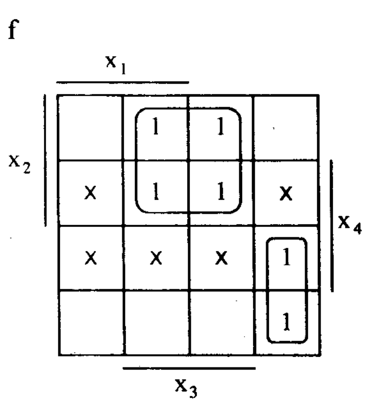
MNO = MDNO

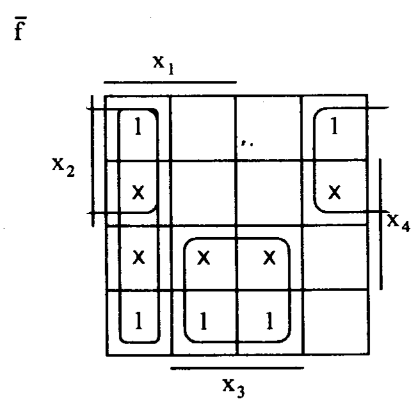
Za MNO je izbrana MDNO, ker je zanjo potrebnih le 10 vhodov v operatorje, kar je manj kot za MKNO.

4.1.4 Minimizacija nepopolnih preklopnih funkcij (funkcije z redundancami)

O nepopolnih preklopnih vezjih govorimo takrat, kadar je na vhodu definiranih manj različnih vhodnih kombinacij, kot je možnih. Redundantna vhodna kombinacija w_i je tista, ki se nikoli ne pojavi na vhodu vezja in ji je zato pripisana vrednost minterma $m_i = 0$. Če velja $m_i = 0$, potem velja tudi $m_i f_i = 0$ neodvisno od funkcijske vrednosti f_i. V postopku minimizacije je lahko i - ti vhodni kombinaciji pripisana takšna funkcijska vrednost f_i kot bolj ustreza minimizacijskemu postopku za iskanje MDNO ali MKNO.

PRIMER 4.8 Za preklopno funkcijo z redundantnimi vhodnimi kombinacijami poiščite MDNO in MKNO.
$f(x_1,x_2,x_3,x_4) = \vee\ (0,1,6,7,14,15)$ in $\vee_X\ (3,5,9,11,13)$ - redundance





MDNO: $f(x_1,x_2,x_3,x_4) = x_2 x_3 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3$

f (NEG): $f(x_1,x_2,x_3,x_4) = x_1 \overline{x}_3 \vee x_2 \overline{x}_3 \vee \overline{x}_2 x_3$

MKNO: $f(x_1,x_2,x_3,x_4) = \overline{x_1 \overline{x}_3 \vee x_2 \overline{x}_3 \vee \overline{x}_2 x_3} = (\overline{x}_1 \vee x_3)(\overline{x}_2 \vee x_3)(x_2 \vee \overline{x}_3)$

NALOGE:

Za preklopne funkcije poiščite MDNO in MKNO ter določite njeno MNO.

1. $f(x_1,x_2,x_3,x_4) = \vee\,(2,3,6,7,8,11,12,14,15)$

R: MDNO: $f(x_1,x_2,x_3,x_4) = \bar{x}_1 x_3 \vee x_3 x_4 \vee x_2 x_3 \vee x_1 \bar{x}_3 \bar{x}_4$

MKNO: $f(x_1,x_2,x_3,x_4) = (x_1 \vee x_3)(x_3 \vee \bar{x}_4)(\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee x_4)$

Določitev MNO: MDNO: (5,13), MKNO: (4,11) $\rightarrow$ MNO = MKNO

2. $f(x_1,x_2,x_3,x_4) = \&\,(13,9,8,7,5,4,3,1)$

R: MDNO: $f(x_1,x_2,x_3,x_4) = \bar{x}_3 x_4 \vee x_1 x_2 x_4 \vee \bar{x}_1 \bar{x}_2 x_4 \vee \bar{x}_1 \bar{x}_3$

MKNO: $f(x_1,x_2,x_3,x_4) = (\bar{x}_1 \vee x_4)(\bar{x}_3 \vee x_4)(\bar{x}_1 \vee x_2 \vee \bar{x}_3)(x_1 \vee \bar{x}_2 \vee \bar{x}_3)$

Določitev MNO: MDNO: (5,14), MKNO: (5,14) $\rightarrow$ MNO = MDNO = MKNO

4.2 Quine-ova metoda minimizacije preklopnih funkcij

Postopek:

1. Funkcijo zapišemo v PDNO (v tabelo vpišemo vse minterme, ki imajo funkcijsko vrednost 1).

2. Poiščemo glavne vsebovalnike tako, da vsak minterm primerjamo z vsakim in če sta minterma sosednji konjunkciji izpustimo spremenljivko, ki omogoča sosednost. Sosedne minterme prečrtamo, ker dajo konjunktivne izraze dolžine n-1. V nadaljevanju upoštevamo za iskanje sosednosti tudi že prečrtane minterme. Izrazi dolžine n-1 so zopet lahko sosedni, zato postopek nadaljujemo vse dokler obstojajo sosedne konjunkcije. Vsi izrazi, ki ostanejo neprečrtani v tabeli, so glavni vsebovalniki.

3. Napravimo tabelo, ki ima toliko vrstic kot je glavnih vsebovalnikov in toliko stolpcev kot je mintermov. V tabeli označimo pripadnost mintermov v glavnih vsebovalnikih.

4. Določimo potrebne glavne vsebovalnike:
stolpec, ki vsebuje en sam znak vsebovanja je pripadajoči glavni vsebovalnik potreben v MDNO. Obkrožimo ga in brišemo iz tabele vse minterme, ki imajo znak vsebovanja pri potrebnem vsebovalniku. To ponovimo za vse kolone z enim samim znakom vsebovanja.

5. Narišemo novo tabelo, ki ima v vsakem stolpcu dva ali več znakov vsebovanja. V primeru enakih stolpcev izpišemo samo enega in preidemo na končno obliko.

6. Med preostalimi glavnimi vsebovalniki v novi tabeli izberemo tiste, ki najugodneje pokrijejo vse preostale minterme, kar pomeni, da upoštevamo dolžino glavnih vsebovalnikov.

PRIMER 4.9 Poiščite MDNO za funkcijo $f(x_1,x_2,x_3,x_4) = \vee\ (0,1,5,6,7,13,14,15)$ s Quine-ovo metodo minimizacije.

V kolono N vpišemo minterme s funkcijsko vrednostjo 1 in poiščemo glavne vsebovalnike.

N	N-1	N-2
$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$ *	$\bar{x}_1\bar{x}_2\bar{x}_3$	x_2x_4
$\bar{x}_1\bar{x}_2\bar{x}_3x_4$ *	$\bar{x}_1\bar{x}_3x_4$	x_2x_3
$\bar{x}_1x_2\bar{x}_3x_4$ *	$\bar{x}_1x_2x_4$ *	
$\bar{x}_1x_2x_3\bar{x}_4$ *	$x_2\bar{x}_3x_4$ *	
$\bar{x}_1x_2x_3x_4$ *	$\bar{x}_1x_2x_3$ *	
$x_1x_2\bar{x}_3x_4$ *	$x_2x_3\bar{x}_4$ *	
$x_1x_2x_3\bar{x}_4$ *	$x_2x_3x_4$ *	
$x_1x_2x_3x_4$ *	$x_1x_2x_4$ *	
	$x_1x_2x_3$ *	

* - konjunktivni izraz je že upoštevan (prečrtan)

V tabelo vpišemo glavne vsebovalnike (konjunktivni izrazi, ki nimajo *).

	0	1	5	6	7	13	14	15
$(\bar{x}_1\bar{x}_2\bar{x}_3)$	√	√						
$\bar{x}_1\bar{x}_3x_4$		√	√					
(x_2x_4)			√		√	√		√
(x_2x_3)				√	√		√	√

Z oklepajem ( ) so označeni potrebni glavni vsebovalniki, za katere ugotovimo, da pokrijejo vse minterme v MDNO.

Kolona pri mintermu 0 ima en sam znak vsebovanja, zato je glavni vsebovalnik $\bar{x}_1\bar{x}_2\bar{x}_3$ potreben in ga označimo ter prečrtamo njegovo vrstico in kolono pri mintermu 0. Isto ponovimo za kolono pri mintermu 6 in mintermu 14. Ko nimamo nobene kolone več z enim samim znakom vsebovanja pogledamo ali smo z označenimi glavnimi vsebovalniki pokrili vse minterme:

glavni vsebovalnik $\bar{x}_1\bar{x}_2\bar{x}_3$ pokrije minterma 0,1

glavni vsebovalnik x_2x_4 pokrije minterme 5,7,13,15 glavni vsebovalnik $x_2 x_3$ pokrije minterme 6,7,14,15

Iz zapisa lahko vidimo, da z zgornjimi tremi glavnimi vsebovalniki pokrijemo vse minterme, zato je minimalna disjunktivna normalna oblika določena z njimi.

MDNO: $f(x_1,x_2,x_3,x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_2 x_4 \vee x_2 x_3$

PRIMER 4.10 Poiščite MDNO za preklopno funkcijo $f(x_1,x_2,x_3) = \vee\ (0,1,3,5,7)$ s Quine-ovo metodo minimizacije.

V kolono N vpišemo minterme s funkcijsko vrednostjo 1 in poiščemo glavne vsebovalnike.

N	N-1	N-2
$\bar{x}_1\bar{x}_2\bar{x}_3$ *	$\bar{x}_1\bar{x}_2$	x_3
$\bar{x}_1\bar{x}_2 x_3$ *	$\bar{x}_1 x_3$ *	
$\bar{x}_1 x_2 x_3$ *	$\bar{x}_2 x_3$ *	
$x_1\bar{x}_2 x_3$ *	$x_2 x_3$ *	
$x_1 x_2 x_3$ *	$x_1 x_3$ *	

* - konjunktivni izraz je že upoštevan (prečrtan)

V tabelo vpišemo glavne vsebovalnike (konjunktivni izrazi, ki nimajo *).

	0	1	3	5	7
$(\bar{x}_1\bar{x}_2)$	√	√			
(x_3)		√	√	√	√

Z oklepajem ( ) so označeni potrebni glavni vsebovalniki, za katere ugotovimo, da pokrijejo vse minterme v MDNO.

Kolona pri mintermu 0 ima en sam znak vsebovanja, zato je glavni vsebovalnik $\bar{x}_1\bar{x}_2$ potreben in ga označimo ter prečrtamo njegovo vrstico in kolono pri mintermu 0. Isto ponovimo za kolono pri mintermu 3 in mintermu 7, kjer vidimo, da je drugi potreben glavni vsebovalnik x_3. Ko nimamo nobene kolone več z enim samim znakom vsebovanja pogledamo ali smo z označenimi glavnimi vsebovalniki pokrili vse minterme:

glavni vsebovalnik $\bar{x}_1\bar{x}_2$ pokrije minterma 0,1

glavni vsebovalnik x_3 pokrije minterme 1,3,5,7

Iz zapisa lahko vidimo, da z zgornjima glavnima vsebovalnikoma pokrijemo vse minterme in minimalna disjunktivna normalna oblika je:

MDNO: $f(x_1,x_2,x_3) = \bar{x}_1\bar{x}_2 \vee x_3$

PRIMER 4.11 Poiščite MDNO za preklopno funkcijo $f(x_1,x_2,x_3) = \vee\,(0,3,4,5,7)$ s Quine-ovo metodo minimizacije.

N	N-1
$\overline{x}_1\overline{x}_2\overline{x}_3$ *	$\overline{x}_2\overline{x}_3$
$\overline{x}_1 x_2 x_3$ *	$x_2 x_3$
$x_1\overline{x}_2\overline{x}_3$ *	$x_1\overline{x}_2$
$x_1\overline{x}_2 x_3$ *	$x_1 x_3$
$x_1 x_2 x_3$ *	

* - konjunktivni izraz je že upoštevan (prečrtan)

	0	3	4	5	7
$(\overline{x}_2\overline{x}_3)$	√		√		
$(x_2 x_3)$		√			√
$x_1\overline{x}_2$			√	√	
$x_1 x_3$				√	√

Z oklepajem ( ) so označeni potrebni glavni vsebovalniki, za katere ugotovimo, da pokrijejo minterme, kjer imamo en znak vsebovanja.

Kolona pri mintermu 0 ima en sam znak vsebovanja, zato je glavni vsebovalnik $\overline{x}_2\overline{x}_3$ potreben in ga označimo ter prečrtamo njegovo vrstico in kolono pri mintermu 0. Isto ponovimo za kolono pri mintermu 3, kjer vidimo, da je drugi potreben glavni vsebovalnik $x_2 x_3$. Ko nimamo nobene kolone več z enim samim znakom vsebovanja pogledamo ali smo z označenimi glavnimi vsebovalniki pokrili vse minterme:

glavni vsebovalnik $\overline{x}_2\overline{x}_3$ pokrije minterma 0,4

glavni vsebovalnik $x_2 x_3$ pokrije minterma 3,7

Iz zapisa lahko vidimo, da z zgornjima glavnima vsebovalnikoma ne pokrijemo vseh mintermov. Ostal nam je minterm 5, ki še ni upoštevan v nobenem glavnem vsebovalniku, ker ima znak vsebovanja pri dveh glavnih vsebovalnikih. Narišemo novo tabelo:

	5
$x_1\overline{x}_2$	√
$x_1 x_3$	√

Minterm 5 pokrijeta oba glavna vsebovalnika in ker sta enake dolžine, je vseeno katerega upoštevamo v MDNO.

MDNO: $f(x_1,x_2,x_3) = \overline{x}_2\overline{x}_3 \vee x_2 x_3 \vee x_1 x_3$ ali

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

4.3 Določanje glavnih vsebovalnikov iz DNO preklopne funkcije

4.3.1 Implikacija in glavni vsebovalniki

Preklopna funkcija f **implicira** funkcijo g, če je povsod tam, kjer je f=1 tudi g=1, ne velja pa nujno obratno. Relacijo implikacije zapišemo

$$f \subseteq g$$

Če obstoja še posebna vhodna kombinacija, ki zadošča f=0 in g=1, potem zapišemo

$$f \subset g$$

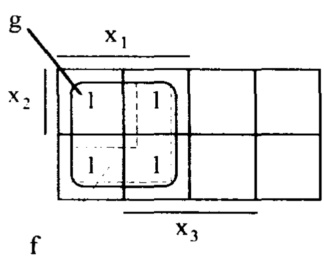
in pravimo, da f **strikno implicira** g.

Ugotavljanje relacije implikacije med f in g z uporabo pravilnostne tabele:

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

V tabeli vidimo, da f implicira g ($f \subseteq g$), ker je povsod tam kjer je f=1 tudi g=1. Istočasno pa velja, da imamo vhodno kombinacijo z f=0 in g=1, zato pravzaprav velja, da f striktno imlicira g ($f \subset g$).

Ugotavljanje relacije implikacije med f in g z uporabo Veitchevega diagrama:



$f \subset g$, $f \subseteq g$

$f \subseteq g$ - f implicira g, če so enice funkcije f znotraj področja enic funkcije g
$f \subset g$ - f striktno implicira g, ker obstoja še kombinacija vhodnih spremenljivk (minterm 6), kjer je f = 0 in g = 1

Vsebovalnik preklopne funkcije f je izraz, ki implicira f.
Izraz implicira f, če velja za vsak x, kjer je izraz = 1, je tudi f(x) = 1, ne velja pa nujno obratno.

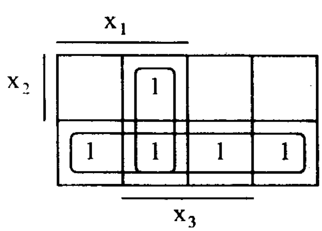
PRIMER 4.12 Poiščite vsebovalnike funkcije $f(x_1,x_2,x_3) = x_1x_3 \vee \overline{x}_2$.

x_1	x_2	x_3	f	x_1x_2	$x_1\overline{x}_2$	$\overline{x}_1\overline{x}_2$	x_1x_3
0	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	1	0	1	0	0
1	0	1	1	0	1	0	1
1	1	0	0	1	0	0	0
1	1	1	1	1	0	0	1

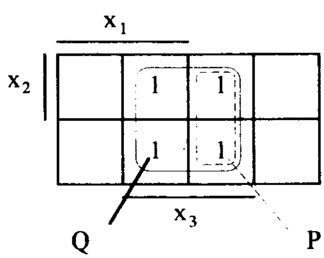
x_1x_2 je izraz, ki ni vsebovalnik funkcije f, ker ne implicira f.
$x_1\overline{x}_2$, $\overline{x}_1\overline{x}_2$, x_1x_3 so vsebovalniki funkcije f, ker vsak izraz implicira f

Glavni vsebovalnik funkcije f je tisti vsebovalnik, ki ni vsebovan v nobenem krajšem izrazu, ki je tudi vsebovalnik funkcije f.

PRIMER 4.13 Vsebovanost in glavni vsebovalniki



Glavni vsebovalniki: $\overline{x}_2$, x_1x_3



P je vsebovan v Q
Q je glavni vsebovalnik

Konsenzus je produkt dveh izrazov P in Q ob pogoju, da se izraza razlikujeta v eni spremenljivki za negacijo oz. nenegacijo spremenljivke. V konsenzusu odpade spremenljivka x_i in $\overline{x}_i$.

$P = x_i.P'$ $\quad$ $Q = \overline{x}_i.Q'$ $\quad$ Konsenzus: $P.Q = P'.Q'$

PRIMER 4.14 Poiščite konsenzus P.Q
$P = x_1\overline{x}_2x_4x_5$ $\quad$ $Q = x_1x_2x_3x_5$ $\quad$ Konsenzus: $P.Q = x_1x_3x_4x_5$

4.3.2 Iterativna metoda določanja glavnih vsebovalnikov

Postopek:
1. Izločimo vse konjunktivne izraze, ki so vsebovani v drugih izrazih.

2. Za poljubna dva izraza, ki tvorita konsenzus formiramo nov zapis funkcije tako, da prvotnemu zapisu funkcije disjunktivno dodamo konsenzus.

Koraka 1. in 2. ponavljamo dokler ni izpolnjen pogoj:
- noben izraz ni vsebovan v drugem,
- poljuben par izrazov, ali nima več konsenzusa ali pa ima konsenzus, ki je vsebovan v nekem izrazu trenutne funkcije

PRIMER 4.15 $f(x_1,x_2,x_3,x_4) = \overline{x}_1 x_2 x_4 \vee x_2 x_3 \overline{x}_4 \vee x_1 x_2 x_4 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3$

2. konsenzus $\overline{x}_1 x_2 x_4$ in $x_1 x_2 x_4 \rightarrow x_2 x_4$

$f(x_1,x_2,x_3,x_4) = \overline{x}_1 x_2 x_4 \vee x_2 x_3 \overline{x}_4 \vee x_1 x_2 x_4 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3 \vee x_2 x_4$

1. izraza $\overline{x}_1 x_2 x_4$ in $x_1 x_2 x_4$ vsebovana v $x_2 x_4$, zato ju izločimo

$f(x_1,x_2,x_3,x_4) = x_2 x_3 \overline{x}_4 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3 \vee x_2 x_4$

2. konsenzus $x_2 x_3 \overline{x}_4$ in $x_2 x_4 \rightarrow x_2 x_3$

$f(x_1,x_2,x_3,x_4) = x_2 x_3 \overline{x}_4 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3 \vee x_2 x_4 \vee x_2 x_3$

1. izraz $x_2 x_3 \overline{x}_4$ vsebovan v $x_2 x_3$, zato ga izločimo

$f(x_1,x_2,x_3,x_4) = \overline{x}_1 \overline{x}_2 \overline{x}_3 \vee x_2 x_4 \vee x_2 x_3$

2. konsenzus $\overline{x}_1 \overline{x}_2 \overline{x}_3$ in $x_2 x_4 \rightarrow \overline{x}_1 \overline{x}_3 x_4$

$f(x_1,x_2,x_3,x_4) = \overline{x}_1 \overline{x}_2 \overline{x}_3 \vee x_2 x_4 \vee x_2 x_3 \vee \overline{x}_1 \overline{x}_3 x_4$

2. konsenzus $\overline{x}_1 \overline{x}_2 \overline{x}_3$ in $x_2 x_4 \rightarrow \overline{x}_1 \overline{x}_3 x_4$ že obstoja, zato ga ne upoštevamo več

2. konsenzus $x_2 x_3$ in $\overline{x}_1 \overline{x}_3 x_4 \rightarrow \overline{x}_1 x_2 x_4$ vsebovan v $x_2 x_4$

Postopek iskanja glavnih vsebovalnikov je končan.

$$f(x_1,x_2,x_3,x_4) = \overline{x}_1 \overline{x}_2 \overline{x}_3 \vee x_2 x_4 \vee x_2 x_3 \vee \overline{x}_1 \overline{x}_3 x_4$$

PRIMER 4.16 $f(x_1,x_2,x_3,x_4) = x_1 \overline{x}_3 x_4 \vee x_1 x_2 x_3 \vee x_1 \overline{x}_2 x_4 \vee x_1 \overline{x}_2 x_3 \overline{x}_4 \vee \overline{x}_1 x_2 \overline{x}_4$

2. konsenzus $x_1 \overline{x}_3 x_4$ in $x_1 x_2 x_3 \rightarrow x_1 x_2 x_4$

$f(x_1,x_2,x_3,x_4) = x_1 \overline{x}_3 x_4 \vee x_1 x_2 x_3 \vee x_1 \overline{x}_2 x_4 \vee x_1 \overline{x}_2 x_3 \overline{x}_4 \vee \overline{x}_1 x_2 \overline{x}_4 \vee x_1 x_2 x_4$

2. konsenzus $x_1x_2x_3$ in $x_1\bar{x}_2x_4 \rightarrow x_1x_3x_4$

$f(x_1,x_2,x_3,x_4) = x_1\bar{x}_3x_4 \vee x_1x_2x_3 \vee x_1\bar{x}_2x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_2x_4 \vee x_1x_3x_4$

2. konsenzus $x_1\bar{x}_3x_4$ in $x_1x_3x_4 \rightarrow x_1x_4$

$f(x_1,x_2,x_3,x_4) = x_1\bar{x}_3x_4 \vee x_1x_2x_3 \vee x_1\bar{x}_2x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_2x_4 \vee$
$x_1x_3x_4 \vee x_1x_4$

1. izrazi $x_1\bar{x}_3x_4$ in $x_1\bar{x}_2x_4$ in $x_1x_2x_4$ in $x_1x_3x_4$ vsebovani v x_1x_4, zato jih izločimo

$f(x_1,x_2,x_3,x_4) = x_1x_2x_3 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_4$

2. konsenzus $x_1x_2x_3$ in $x_1\bar{x}_2x_3\bar{x}_4 \rightarrow x_1x_3\bar{x}_4$

$f(x_1,x_2,x_3,x_4) = x_1x_2x_3 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_4 \vee x_1x_3\bar{x}_4$

1. izraz $x_1\bar{x}_2x_3\bar{x}_4$ vsebovan v $x_1x_3\bar{x}_4$, zato ga izločimo

$f(x_1,x_2,x_3,x_4) = x_1x_2x_3 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_4 \vee x_1x_3\bar{x}_4$

2. konsenzus $x_1x_2x_3$ in $\bar{x}_1x_2\bar{x}_4 \rightarrow x_2x_3\bar{x}_4$

$f(x_1,x_2,x_3,x_4) = x_1x_2x_3 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_4 \vee x_1x_3\bar{x}_4 \vee x_2x_3\bar{x}_4$

2. konsenzus x_1x_4 in $x_1x_3\bar{x}_4 \rightarrow x_1x_3$

$f(x_1,x_2,x_3,x_4) = x_1x_2x_3 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_4 \vee x_1x_3\bar{x}_4 \vee x_2x_3\bar{x}_4 \vee x_1x_3$

1. izraza $x_1x_2x_3$ in $x_1x_3\bar{x}_4$ vsebovana v x_1x_3, zato ju izločimo

$f(x_1,x_2,x_3,x_4) = \bar{x}_1x_2\bar{x}_4 \vee x_1x_4 \vee x_2x_3\bar{x}_4 \vee x_1x_3$

2. konsenzus $\bar{x}_1x_2\bar{x}_4$ in $x_1x_3 \rightarrow x_2x_3\bar{x}_4$ že obstoja
2. konsenzus x_1x_4 in $x_2x_3\bar{x}_4 \rightarrow x_1x_2x_3$ vsebovan v x_1x_3

Postopek iskanja glavnih vsebovalnikov je končan.

$$f(x_1,x_2,x_3,x_4) = \bar{x}_1x_2\bar{x}_4 \vee x_1x_4 \vee x_2x_3\bar{x}_4 \vee x_1x_3$$

NALOGE:

Določite glavne vsebovalnike z iterativno metodo iskanja:

1. $f(x_1,x_2,x_3,x_4) = \bar{x}_1x_3 \vee x_1\bar{x}_2\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_4 \vee \bar{x}_2x_3x_4 \vee x_1x_2\bar{x}_4$

R: $f(x_1,x_2,x_3,x_4) = \bar{x}_1x_3 \vee \bar{x}_2x_3 \vee x_2\bar{x}_4 \vee x_1\bar{x}_4 \vee x_3\bar{x}_4$

2. $f(x_1,x_2,x_3,x_4) = \bar{x}_1x_2\bar{x}_3 \vee x_1x_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_4 \vee x_1x_3x_4 \vee \bar{x}_1x_2\bar{x}_4$

R: $f(x_1,x_2,x_3,x_4) = \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_4 \vee x_2\bar{x}_3x_4$

4.4 Izračun vseh neredundatnih disjunktivnih oblik preklopne funkcije (Tison-ova metoda)

Pri iskanju neredundatne disjunktivne oblike (minimalne oblike) z uporabo Tison-ove metode izhajamo iz glavnih vsebovalnikov, ki smo jih dobili z iterativno metodo iskanja.

Postopek:

1. Izdelajmo formacijsko drevo posplošenega konsenzusa:

- Glavne vsebovalnike zapišemo v vrsto in jih označimo z indeksi (A, B, C, ...).

- Izberemo prvo dvolično spremenljivko (npr. x_i).

- Generiramo vse možne konsenzuse za dvolično spremenljivko x_i, ter jih zapišemo v novo vrstico in jih označimo s črto kako pridemo do njih ter s produktom indeksov iz katerih je konsenzus. Za vsak novi produkt v isti vrstici ali višje pogledamo, če vsebuje obstoječe produkte, ali je vsebovan v njih, pri čemer se indeksi tretirajo kot običajni literali in potem prečrtamo produkt, ki je vsebovan v drugem. Če nov produkt ni glavni vsebovalnik ga podčrtamo.

- Izberemo naslednjo dvolično spremenljivko in ponovimo 1.c

V najnižji vrstici ni potrebno pisati produktov, ki niso glavni vsebovalniki funkcije f oz. so vsebovani v drugih produktih.

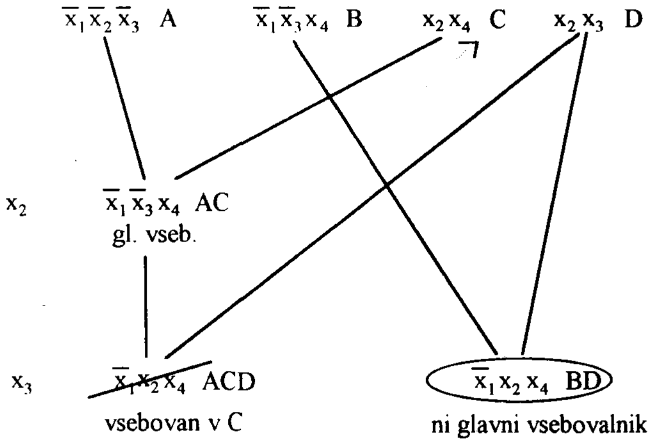
2. Formiramo vsebovane funkcije v formacijskem drevesu tako, da za vsak glavni vsebovalnik npr. P poiščemo vse indeksne produkte, ki imajo isti P in nato poiščemo vsebovano funkcijo tako, da disjunktivno povežemo vse indeksne produkte.

3. Določimo Tison-ovo funkcijo tako, da konjunktivno povežemo vsebovane funkcije in jih poenostavimo s postulati in izreki.

4. Zapišemo neredundantno disjunktivno normalno obliko funkcije tako, da disjunktivno povežemo vse glavne vsebovalnike, ki smo jih dobili v Tison-ovi funkciji.

PRIMER 4.17 Zapišite neredundatno disjunktivno normalno obliko za preklopno funkcijo $f(x_1,x_2,x_3,x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_3x_4 \vee x_2x_4 \vee x_2x_3$

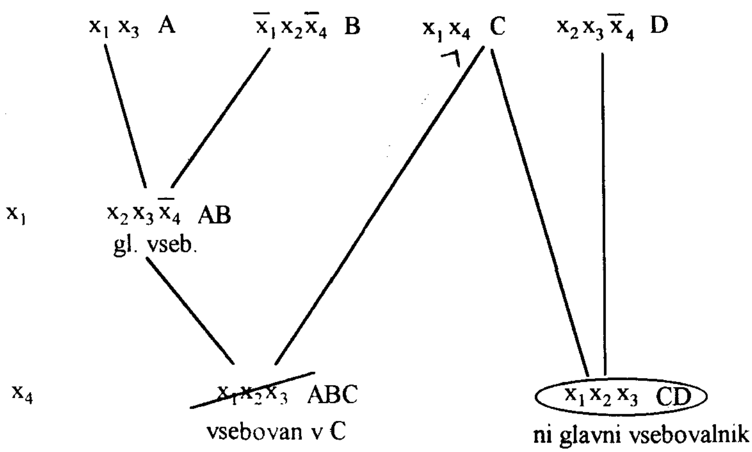
1. Izdelamo formacijsko drevo posplošenega konsenzusa



2. Vsebovane funkcije so: A , (B V AC) , C , D
3. Tison-ova funkcija: A.C.(B V AC).D = A.C.D (izrek: X(Y V X) = X)
4. Neredundantna disjunktivna oblika: $f(x_1,x_2,x_3,x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_2x_4 \vee x_2x_3$

PRIMER 4.18 Zapišite neredundatno disjunktivno normalno obliko za preklopno funkcijo $f(x_1,x_2,x_3,x_4) = x_1x_3 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_4 \vee x_2x_3\bar{x}_4$

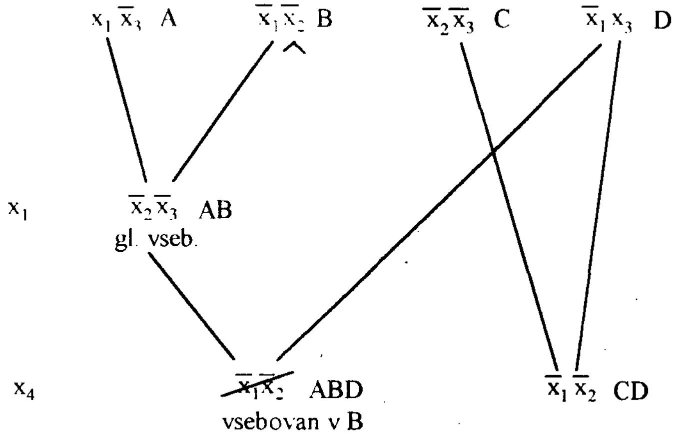
1. Izdelamo formacijsko drevo posplošenega konsenzusa



2. Vsebovane funkcije so: A , B , C , D v AB
3. Tison-ova funkcija: A.B.C.(D v AB) = A.B.C.D v A.B.C = A.B.C
4. Neredundantna disjunktivna oblika: $f(x_1,x_2,x_3,x_4) = x_1x_3 \vee \bar{x}_1x_2\bar{x}_4 \vee x_1x_4$

PRIMER 4.19 Zapišite neredundatno disjunktivno normalno obliko za preklopno funkcijo $f(x_1,x_2,x_3) = x_1\bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_2\bar{x}_3 \vee \bar{x}_1 x_3$

1. Izdelamo formacijsko drevo posplošenega konsenzusa



2. Vsebovane funkcije so: A , B v CD , C v AB , D
3. Tison-ova funkcija:

A.(B v C.D) .(C v A.B) D = (A.B v A.C.D) (C.D v A.B.D) =
=A.B.C.D v A.C.D v A.B.D v A.B.C.D = A.C.D v A.B.D

Tison-ova funkcija nam da dve različni neredundatni disjunktivni obliki ACD in ABD. Za rešitev izberemo eno od njih, ker sta ti dve rešitvi enakovredni.

4. Neredundatni disjunktivni obliki:

a. $f(x_1,x_2,x_3) = x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3 \vee \bar{x}_1 x_3$

b. $f(x_1,x_2,x_3) = x_1\bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1 x_3$

NALOGE:

Poiščite MDNO za podane preklopne funkcije s Tison-ovo metodo

1. $f(x_1,x_2,x_3,x_4) = x_3 x_4 \vee \bar{x}_2 x_4 \vee \bar{x}_2\bar{x}_3$

R: MDNO: $f(x_1,x_2,x_3,x_4) = x_3 x_4 \vee \bar{x}_2\bar{x}_3$

2. $f(x_1,x_2,x_3,x_4) = \bar{x}_1 x_2 \bar{x}_4 \vee x_1 x_3 \vee x_2 x_3 \bar{x}_4 \vee x_1 x_4$

R: MDNO: $f(x_1,x_2,x_3,x_4) = \bar{x}_1 x_2 \bar{x}_4 \vee x_1 x_3 \vee x_1 x_4$

5 PREKLOPNE FUNKCIJE S POSEBNIMI LASTNOSTMI

5.1 Simetrične preklopne funkcije

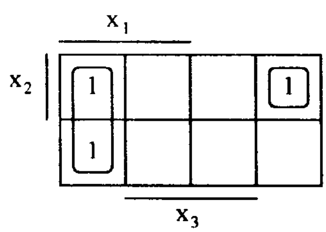
Preklopna funkcija je delno simetrična za spremenljivki x_i in x_j, če je preklopna funkcija nespremenjena pri zamenjavi i-te spremenljivke z j-to spremenljivko.

$$x_i \sim x_j:\ f(x_1,x_2\ldots,x_i,\ldots,x_j,\ldots,x_n) = f(x_1,x_2\ldots,x_j,\ldots,x_i,\ldots,x_n)$$

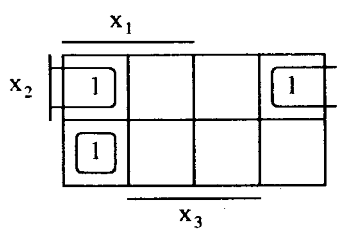
PRIMER 5.1 Delno simetrična funkcija - $f(x_1,x_2,x_3) = x_1\bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3$

$x_1 \sim x_2$ - zamenjamo x_1 in x_2 v preklopni funkciji
$f(x_1,x_2,x_3) = x_2\bar{x}_3 \vee \bar{x}_2 x_1 \bar{x}_3 = x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3$

V Veitchevem diagramu vidimo enakost obeh preklopnih funkcij



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



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

Preklopna funkcija je popolnoma simetrična, če velja za vsak par (x_i, x_j) zamenjava $x_i \sim x_j$ ali $x_i \sim \bar{x}_j$:

$$x_i \sim x_j:\quad f(x_1,\ldots,x_i,\ldots,x_j,\ldots,x_n) = f(x_1,\ldots,x_j,\ldots,x_i,\ldots,x_n)$$

$$x_i \sim \bar{x}_j:\quad f(x_1,\ldots,x_i,\ldots,x_j,\ldots,x_n) = f(x_1,\ldots,\bar{x}_j,\ldots,\bar{x}_i,\ldots,x_n)$$

Potreben in zadosten pogoj, da je preklopna funkcija simetrična je obstoj množice števil $A = (a_1,a_2,\ldots,a_k)$, $0 \le a_i \le n$ tako, da je pri a_i-vhodnih spremenljivkah z vrednostjo 1, vrednost funkcije le v tem primeru 1. Če funkcija zadošča tej definiciji jo lahko zapišemo

$$f(x_1,x_2,\ldots,x_n) = f_A(x_1^{w_1},x_2^{w_2},\ldots,x_n^{w_n})$$

A - simetrijska množica
$x_1^{w_1},x_2^{w_2},\ldots,x_n^{w_n}$ - simetrijski nabor neodvisnih spremenljivk

5.1.1 Ugotavljanje simetričnosti

Analitičen dokaz simetričnosti

Postopek:
1. Zapišemo v tabelo vse vhodne kombinacije s funkcijsko vrednostjo 1.

2. Pogledamo enakost ali recipročnost števila ničel in števila enic vstolpcih (potreben pogoj):

- enakost je izpolnjena (simetrijski nabor neodvisnih spremenljivk je kar osnovni nabor spremenljivk)
- enakost do recipročnosti - z negacijo ustrezne spremenljivke (stolpca) je možno dobiti enakost ničel in enic (simetrijski nabor neodvisnih spremenljivk je definiran tudi z negiranimi spremenljivkami).

3. Zapišemo število enic v vrsticah tabele (določanje utežnega vektorja).

4. Preverimo zadosten pogoj simetričnosti - število vrstic v tabeli z utežjo u mora biti enako

$$_{n}V_{u} = \frac{n!}{(n-u)!u!}$$

PRIMER 5.2 Ali je funkcija $f(x_1,x_2,x_3,x_4) = v\ (1,2,4,7,8,11,13,14)$ simetrična.

x_1	x_2	x_3	x_4	u	
0	0	0	1	1	
0	0	1	0	1	
0	1	0	0	1	
0	1	1	1	3	
1	0	0	0	1	
1	0	1	1	3	
1	1	0	1	3	
1	1	1	0	3	
4/4	4/4	4/4	4/4		št. ničel / št. enic

u - utežni vektor

Potreben pogoj je izpolnjen, ker je število ničel in število enic enako v vseh stolpcih.

Zadosten pogoj:
Izračunamo število vhodnih kombinacij za u =1,3.

$$ {}_4v_1 = \frac{4!}{3!\ 1!} = 4 \qquad {}_4v_3 = \frac{4!}{1!\ 3!} = 4 $$

V tabeli imamo štiri vhodne kombinacije z eno enico in štiri vhodne kombinacije s tremi enicami, zato je tudi zadosten pogoj izpolnjen.

Simetrična preklopna funkcija: $f(x_1,x_2,x_3,x_4) = f_{(1,3)}(x_1,x_2,x_3,x_4)$

PRIMER 5.3 Ali je preklopna funkcija $f(x_1,x_2,x_3) = \vee\ (1,2,4,5,7)$ simetrična.

x_1	x_2	x_3	x_1	$\overline{x}_2$	x_3	u
0	0	1	0	1	1	2
0	1	0	0	0	0	0
1	0	0	1	1	0	2
1	0	1	1	1	1	3
1	1	1	1	0	1	2
2 /3	3 / 2	2 / 3	2 / 3	2 / 3	2 / 3	

Potreben pogoj smo izpolnili z negacijo spremenljivke x_2

Zadosten pogoj:

$$ {}_3v_2 = \frac{3!}{1!\ 2!} = 3 \qquad {}_3v_3 = \frac{3!}{0!\ 3!} = 1 \qquad {}_3v_0 = \frac{3!}{3!\ 0!} = 1 $$

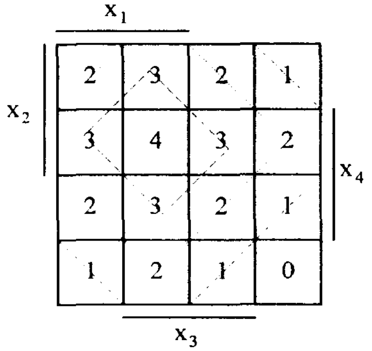
V tabeli imamo tri vhodne kombinacije z eno enico in po eno vhodno kombinacijo z nobeno enico in tremi enicami, zato je tudi zadosten pogoj izpolnjen.

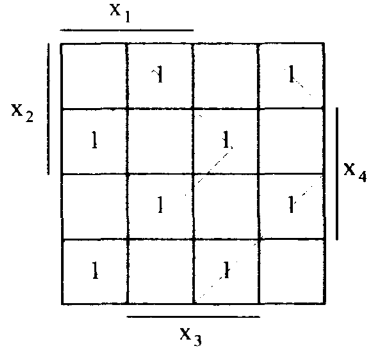
Simetrična preklopna funkcija: $f(x_1,x_2,x_3) = f_{(0,2,3)}(x_1,\overline{x}_2,x_3)$

Grafičen dokaz simetričnosti

Podana preklopna funkcija je simetrična, če najdemo v Veitchevem diagramu sovpadanje funkcijskih vrednosti 1 z vsemi utežmi, ki definirajo simetrijsko množico. Ta dokaz je enostaven, če je funkcija simetrična za osnovni simetrijski nabor, sicer moramo ta postopek iskanja simetričnosti vpeljati na razširjenem Veitchevem diagramu. Pregledati moramo vse možne nabore za ugotavljanje simetričnosti ali nesimetričnosti funkcije.

PRIMER 5.4 Ali je funkcija $f(x_1,x_2,x_3,x_4) = \vee(1,2,4,7,8,11,13,14)$ simetrična.

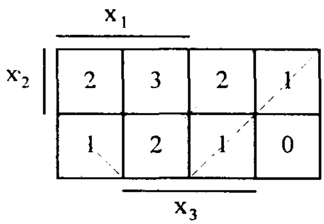


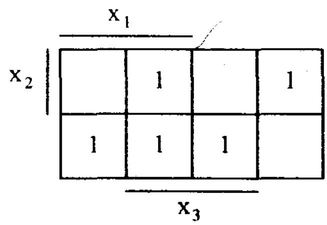


Zgornja preklopna funkcijo ima v Veitchevem diagramu funkcijske vrednosti 1 povsod tam, kjer so uteži 1 in 3, kar pomeni, da je simetrijska množica A = (1,3) in simetrijski nabor (x_1,x_2,x_3,x_4).

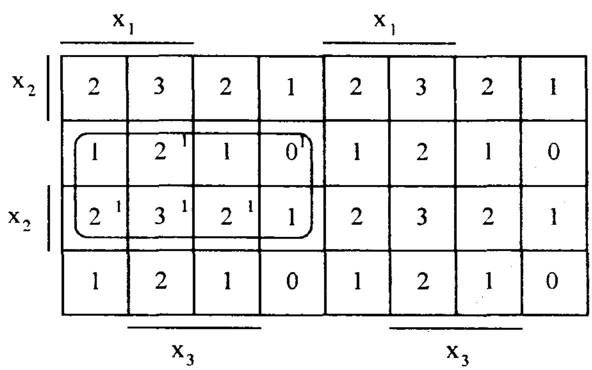
Simetrična preklopna funkcija: $f(x_1,x_2,x_3,x_4) = f_{(1,3)}(x_1,x_2,x_3,x_4)$

PRIMER 5.5 Ali je preklopna funkcija $f(x_1,x_2,x_3) = \vee(1,2,4,5,7)$ simetrična.





V diagramu vidimo, da ni izpolnjen pogoj simetričnosti za osnovni simetrijski nabor, ker nimamo enice pri mintermu 6, zato moramo določiti razširjeni Veitchev diagram.



Pri negirani spremenljivki $\bar{x}_2$ (premaknili smo diagram s funkcijskimi vrednostmi 1 navzdol v razširjenem diagramu) imamo izpolnjen pogoj simetričnosti.

Simetrična preklopna funkcija: $f(x_1,x_2,x_3) = f_{(0,2,3)}(x_1,\bar{x}_2,x_3)$

5.1.2 Zapis PDNO za podano simetrično funkcijo

Postopek:

1. Določimo število vhodnih kombinacij za vsako utež u v simetrijski množici.

$$ {}_{n}v_{u} = \frac{n!}{(n-u)!\,u!} $$

2. Zapišemo vse možne vhodne kombinacije pri posameznih utežeh za podan simetrijski nabor spremenljivk. Če imamo v simetrijskem naboru negirano spremenljivko, moramo v tabeli ustrezno kolono negirati in poiščemo PDNO za osnovni nabor neodvisnih spremenljivk pri preklopni funkciji.

PRIMER 5.6 Zapišite PDNO za simetrično preklopno funkcijo $f_{(2,4)}(\overline{x}_1, x_2, \overline{x}_3, \overline{x}_4)$

$$ {}_{4}v_{2} = \frac{4!}{2!\,2!} = 6 \qquad {}_{4}v_{4} = \frac{4!}{0!\,4!} = 1 $$

u	$\overline{x}_1$	x_2	$\overline{x}_3$	$\overline{x}_4$	x_1	x_2	x_3	x_4	m
2	0	0	1	1	1	0	0	0	8
2	0	1	0	1	1	1	1	0	14
2	0	1	1	0	1	1	0	1	13
2	1	0	0	1	0	0	1	0	2
2	1	0	1	0	0	0	0	1	1
2	1	1	0	0	0	1	1	1	7
4	1	1	1	1	0	1	0	0	4
	3/4	3/4	3/4	3/4					

Negirali smo koloni pri spremenljivkah x_1, x_3, x_4, da smo dobili osnoven nabor spremenljivk preklopne funkcije, pri katerem določimo minterme s funkcijko vrednostjo 1, ki so zapisani v zadnji koloni tabele.

PDNO: $f(x_1, x_2, x_3, x_4) = \vee\,(1,2,4,7,8,13,14)$

NALOGE:

1. $f_{(2,3)}(x_1, x_2, x_3)$ **R:** $f(x_1, x_2, x_3) = \vee\,(3,5,6,7)$
2. $f_{(0,1,2)}(\overline{x}_1, x_2, x_3, \overline{x}_4)$ **R:** $f(x_1, x_2, x_3, x_4) = \vee\,(0,1,3,5,8,9,10,11,12,13,15)$
3. $f_{(2,4)}(\overline{x}_1, x_2, \overline{x}_3, x_4)$ **R:** $f(x_1, x_2, x_3, x_4) = \vee\,(0,3,5,6,9,12,15)$

5.1.3 Lastnosti simetričnih funkcij

1. **Simetrični polinom** je tista simetrična funkcija, ki v simetrijski množici vsebuje vsa števila od nekega d naprej do n.

$A = (d, d+1, \dots, n)$

2. **Negacija simetrične funkcije** je simetrična funkcija.

$\bar{f}_A(x_1^{w_1}, x_2^{w_2}, \dots, x_n^{w_n}) = f_B(x_1^{w_1}, x_2^{w_2}, \dots, x_n^{w_n})$
$B = \bar{A}$ - pri negaciji funkcije so v množici B manjkajoči elementi množice A

3. **Negacija simetrijskega nabora** je simetrična funkcija.

$f_A(x_1^{\bar{w}_1}, x_2^{\bar{w}_2}, \dots, x_n^{\bar{w}_n}) = f_B(x_1^{w_1}, x_2^{w_2}, \dots, x_n^{w_n})$
$b_i = n - a_i$ - so elementi množice B pri negaciji spremenljivk

4. **Dualna simetrična funkcija** je simetrična funkcija.

$\bar{f}_A(x_1^{\bar{w}_1}, x_2^{\bar{w}_2}, \dots, x_n^{\bar{w}_n}) = f_B(x_1^{w_1}, x_2^{w_2}, \dots, x_n^{w_n})$
$b_i = n - \bar{a}_i$ - so elementi množice B pri dualni simetrični funkciji

5. **Konjunkcija simetričnih funkcij** je simetrična funkcija (simetrijski nabor enak v obeh funkcijah).

$f_C = f_A \& f_B \qquad C = A \cap B$

6. **Disjunkcija simetričnih funkcij** je simetrična funkcija (simetrijski nabor enak v obeh funkcijah).

$f_C = f_A \vee f_B \qquad C = A \cup B$

PRIMER 5.7 Za podano preklopno funkcijo določite simetrijsko množico in simetrijski nabor spremenljivk.

$f(x_1, x_2, x_3) = f_{(2)}(\bar{x}_1, x_2, x_3) \vee \bar{f}_{(0,2)}(\bar{x}_1, \bar{x}_2, \bar{x}_3)$

Rešitev:
1. Poiščemo dualno funkcijo:
$A = (0,2), \quad \bar{A} = (1,3), \quad b_i = n - \bar{a}_i, \quad B = (0,2)$

$\bar{f}_{(0,2)}(\bar{x}_1, \bar{x}_2, \bar{x}_3) = f_{(0,2)}(x_1, x_2, x_3)$

2. Poiščemo simetrično funkcijo za simetrijski nabor $(\bar{x}_1, x_2, x_3)$, ker smemo disjunkcija izvesti le nad enakim simetrijskim naborom za obe funkciji.

$f_{(0,2)}(x_1,x_2,x_3) = f_A(\bar{x}_1,x_2,x_3)$ A = ?

${}_3V_0 = \frac{3!}{3!\,0!} = 1$ ${}_3V_2 = \frac{3!}{1!\,2!} = 3$

x_1	x_2	x_3	$\bar{x}_1$	x_2	x_3	u
0	0	0	1	0	0	1
0	1	1	1	1	1	3
1	0	1	0	0	1	1
1	1	0	0	1	0	1
2 /2	2 / 2	2 / 2	2 / 2	2 / 2	2 / 2	

A = (1,3) - simetrijska množica $f_{(1,3)}(\bar{x}_1,x_2,x_3)$

Simetrično funkcijo $f_{(0,2)}(x_1,x_2,x_3)$ lahko pretvorimo v $f_{(1,3)}(\bar{x}_1,x_2,x_3)$.

3. Rešimo disjunkcijo simetričnih funkcij, kjer je simetrijska množica A določena z unijo obeh simetrijskih množic.

$f(x_1,x_2,x_3) = f_{(2)}(\bar{x}_1,x_2,x_3) \vee f_{(1,3)}(\bar{x}_1,x_2,x_3) = f_{(1,2,3)}(\bar{x}_1,x_2,x_3)$

NALOGE:

1. $f(x_1,x_2,x_3,x_4) = f_{(0,1)}(\bar{x}_1,x_2,x_3,\bar{x}_4) \vee f_{(2)}(x_1,\bar{x}_2,\bar{x}_3,x_4)$

R: $f_{(0,1,2)}(\bar{x}_1,x_2,x_3,\bar{x}_4)$

2. $f(x_1,x_2,x_3,x_4) = f_{(1)}(\bar{x}_1,x_2,\bar{x}_3,x_4) \vee f_{(0,3,4)}(x_1,\bar{x}_2,x_3,\bar{x}_4) \mathbin{\&} \bar{f}_{(0,2)}(x_1,\bar{x}_2,x_3,\bar{x}_4)$

R: $f_{(3,4)}(x_1,\bar{x}_2,x_3,\bar{x}_4)$

3. $f(x_1,x_2,x_3) = f_{(0,3)}(x_1,\bar{x}_2,x_3) \vee \bar{f}_{(1,3)}(x_1,\bar{x}_2,x_3) \mathbin{\&} f_{(1,3)}(\bar{x}_1,x_2,\bar{x}_3)$

R: $f_{(0,2,3)}(x_1,\bar{x}_2,x_3)$

4. $f(x_1,x_2,x_3,x_4) = f_{(3,4)}(x_1,\bar{x}_2,x_3,x_4) \mathbin{\&} \bar{f}_{(0,1,4)}(x_1,\bar{x}_2,x_3,x_4) \vee f_{(3)}(\bar{x}_1,x_2,\bar{x}_3,\bar{x}_4)$

R: $f_{(1,3)}(x_1,\bar{x}_2,x_3,x_4)$

5.2 Pragovne preklopne funkcije

Preklopna funkcija $f(x_1,x_2,...,x_n)$ je linearno ločljiva, če obstoja hiperravnina v Evklidskem prostoru, ki loči funkcijske vrednosti $f(W_i) = 0$ od vrednosti $f(W_i) = 1$. Enačba hiperravnine je podana z

$$a_1x_1 + a_2x_2 + ... + a_nx_n = P$$

$a_1, a_2,...,a_n$ - niz števil, ki ga imenujemo uteži
P - število, ki ga imenujemo prag

Sistem števil $(a_1,a_2,...,a_n;\ P)$ imenujemo sistem linearne ločljivosti, ustrezno preklopno funkcijo, ki zadošča temu sistemu pa pragovna funkcija. Definirana je z

$$f(x_1 x_2,...,x_n) \begin{cases} 1, \text{ če } & a_1x_1 + a_2x_2 + ... + a_nx_n > P \\ 0, \text{ če } & a_1x_1 + a_2x_2 + ... + a_nx_n < P \end{cases}$$

5.2.1 Linearno ločljive funkcije (pragovne funkcije)

Preklopna funkcija je linearno ločljiva (pragovna), če izpolnjuje naslednje pogoje:
- pogoj enotipnosti
- pogoj monotonosti
- pogoj neseštevljivosti

Lastnost linearno ločljivih funkcij je, da jih je možno realizirati z enim samim pragovnim operatorjem, zato jih imenujemo tudi pragovne funkcije.

Pogoj enotipnosti preklopne funkcije

Preklopna funkcija je enotipna, če je v minimalni obliki zapisana tako, da je vsaka spremenljivka uporabljena le v eni obliki (x_i ali $\overline{x}_i$). Pragovne funkcije so podmnožica enotipnih funkcij.

PRIMER 5.8
Preklopni funkciji sta enotipni, ker se vsaka spremenljivka pojavi v eni sami obliki.

$f(x_1,x_2,x_3,x_4) = x_1x_2 \vee x_2x_3$

$f(x_1,x_2,x_3,x_4) = x_1x_2 \vee x_2\overline{x}_3\overline{x}_4 \vee x_1\overline{x}_4$

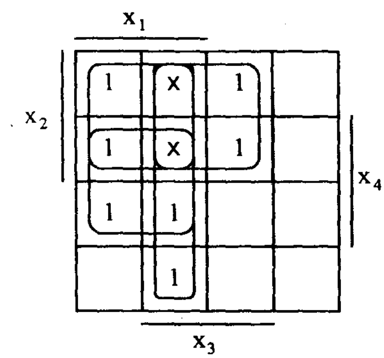
Preklopni funkciji nista enotipni, ker se spremenljivke pojavljajo v obeh oblikah.

$f(x_1,x_2,x_3,x_4) = x_1x_2 \vee x_2x_3 \vee \overline{x}_2x_4$

$f(x_1,x_2,x_3,x_4) = x_1x_2 \vee x_2\overline{x}_3\overline{x}_4 \vee x_1\overline{x}_4 \vee \overline{x}_1x_3x_4$

PRIMER 5.9 Ugotovite za katere vrednosti redundanc × je preklopna funkcija $f(x_1,x_2,x_3,x_4) = \&(15,14,13,12,11,10,7)$ in $\vee_\times$ (14,15) enotipna:

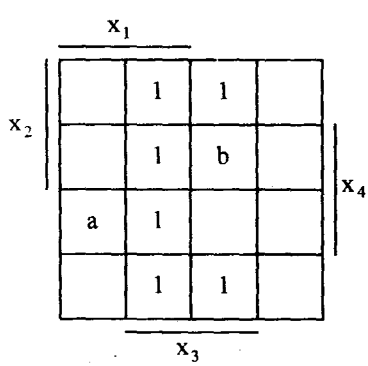
Preklopna funkcija je podana z makstermi pri katerih ima funkcija vrednosti 0 in z mintermoma 14 in15, kjer ima funkcija nedoločeno vrednost (× - redundanca, kjer lahko funkcija zavzame vrednost 0 ali 1).



Preklopna funkcija je enotipna za redundance × = 1. Pri katerikoli drugi izbiri redundanc (00,01,10) funkcija ni enotipna.

MDNO: $f(x_1,x_2,x_3,x_4) = x_1x_2 \vee x_2x_3 \vee x_1x_3 \vee x_1x_4$

PRIMER 5.10 Določite redundanci a, b tako, da bo preklopna funkcija enotipna



Zapišimo MDNO za vse kombinacije vrednosti, ki jih lahko zavzameta a in b:

<u>a b</u>

0 0 $f = x_1x_3 \vee x_3\bar{x}_4$ - enotipna

0 1 $f = x_1x_3 \vee x_2x_3 \vee x_3\bar{x}_4$ - enotipna

1 0 $f = x_1x_3 \vee x_3\bar{x}_4 \vee x_1\bar{x}_2x_4$ - ni enotipna

1 1 $f = x_1x_3 \vee x_2x_3 \vee x_3\bar{x}_4 \vee x_1\bar{x}_2x_4$ - ni enotipna

Preklopna funkcija je enotipna pri: a = b = 0 in a = 0, b = 1.

Pogoj monotonosti preklopne funkcije

Funkcija f(X) je 1-monotona, če sta primerljiva funkcijska ostanka dolžine n-1 pri razčlenitvi preklopne funkcije po spremenljivki x_i, i = 1, 2, ..., n.
Funkcija f(X) je k-monotona, če velja primerljivost za vse možne pare v nizu funkcijskih ostankov dolžine n-k, kjer je preklopna funkcija razčlenjena za vse kombinacije k spremenljivk.
Funkcija f(X) izpolnjuje pogoj monotonosti, če je popolnoma monotona, kar pomeni da je k-monotona za k = 1, 2 ,..., n-1.

Dve funkciji iz niza f(X) in g(X) sta primerljivi, če velja relacija $f(X) \supseteq g(X)$ ali $g(X) \supseteq f(X)$, kjer je $\supseteq$ znak pripadnosti, ki je definiran z izrazom:

$f(X) = 1 \rightarrow g(X) = 1 \leftrightarrow g(X) \supseteq f(X)$ (kjer je f(X) =1, mora biti tudi g(X) = 1)
$g(X) = 1 \rightarrow f(X) = 1 \leftrightarrow f(X) \supseteq g(X)$ (kjer je g(X) =1, mora biti tudi f(X) = 1)

Za $n \leq 8$ je pogoj monotonosti skupaj z enotipnostjo že zadosten pogoj za linearno ločljive funkcije.

PRIMER 5.11 Enostaven primer primerljivosti sta OR in AND funkciji.

Če je $f(X) = x_1 \vee x_2 \vee x_3$ in $g(X) = x_1x_2x_3$, potem velja g(X) = 1 povsod tam, kjer je tudi f(X) = 1, kar zapišemo $f(X) \supseteq g(X)$. Primerljivost opazujemo v Veitchevem diagramu.

g(X) x_1

x_2		1		

x_3

f(X) x_1

x_2	1	1	1	1
	1	1	1	

x_3

PRIMER 5.12 Primerljivost enostavnih funkcij $f_1 = 0$, $f_2 = \overline{x}_1$, $f_3 = 1$.

f_1 x_1

x_2		

f_2 x_1

x_2		1
		1

f_3 x_1

x_2	1	1
	1	1

Vse tri funkcije so primerljive: $f_2 \supseteq f_1$, $f_3 \supseteq f_1$, $f_3 \supseteq f_2$

PRIMER 5.13 Ugotovite ali preklopna funkcija $f(x_1,x_2,x_3) = x_1x_2 \vee x_2x_3$ zadošča pogoju monotonosti.

Za preklopno funkcijo moramo ugotoviti ali je 1- monotona in 2-monotona.

1. Razčlenimo preklopno funkcijo po k = 1 za vseh n spremenljivk - x_1,x_2,x_3. Opazujmo primerjavo funkcij v Veitchevem diagramu.

Razčlenitev po x_1:

$$f(x_1,x_2,x_3) = \overline{x}_1 f_0 \vee x_1 f_1 = \overline{x}_1(x_2x_3) \vee x_1(x_2)$$

Funkcijska ostanka : f_0

	x_2	
x_3	1	

f_1

	x_2	
x_3	1	
	1	

Primerjamo f_0 z f_1 $\rightarrow (f_1 = x_2) \supseteq (f_0 = x_2x_3)$

Razčlenitev po x_2:

$$f(x_1,x_2,x_3) = \overline{x}_2 f_0 \vee x_2 f_1 = \overline{x}_2(0) \vee x_2(x_1 \vee x_3)$$

Funkcijska ostanka : f_0

	x_1	
x_3		

f_1

	x_1	
x_3	1	1
	1	

Primerjamo f_0 z f_1 $\rightarrow (f_1 = x_1 \vee x_3) \supseteq (f_0 = 0)$

Razčlenitev po x_3:

$$f(x_1,x_2,x_3) = \overline{x}_3 f_0 \vee x_3 f_1 = \overline{x}_3(x_1x_2) \vee x_3(x_2)$$

Funkcijska ostanka : f_0

	x_1	
x_2	1	

f_1

	x_1	
x_2	1	1

Primerjamo f_0 z f_1 $\rightarrow (f_1 = x_2) \supseteq (f_0 = x_1x_2)$

Preklopna funkcija je 1 - monotona

2. Razčlenimo preklopno funkcijo po k = n - 1 = 2, kjer imamo funkcijo razčlenjeno po 2-spremenljivkah za vse možne kombinacije: x_1x_2, x_1x_3, x_2x_3.

Razčlenitev po x_1x_2:

$f(x_1,x_2,x_3) = \bar{x}_1\bar{x}_2f_0 \vee \bar{x}_1x_2f_1 \vee x_1\bar{x}_2f_2 \vee x_1x_2f_3$

$f(x_1,x_2,x_3) = \bar{x}_1\bar{x}_2(0) \vee \bar{x}_1x_2(x_3) \vee x_1\bar{x}_2(0) \vee x_1x_2(1)$

Funkcijski ostanki : $f_0 = 0,\ f_1 = x_3,\ f_2 = 0,\ f_3 = 1$

Primerjamo f_0 z $f_1f_2f_3$ $\quad f_1 \supseteq f_0$; $f_2 = f_0$; $f_3 \supseteq f_0$

Primerjamo f_1 z f_2f_3 $\quad f_1 \supseteq f_2$; $f_3 \supseteq f_1$

Primerjamo f_2 z f_3 $\quad f_3 \supseteq f_2$

Razčlenitev po x_1x_3:

$f(x_1,x_2,x_3) = \bar{x}_1\bar{x}_3f_0 \vee \bar{x}_1x_3f_1 \vee x_1\bar{x}_3f_2 \vee x_1x_3f_3$

$f(x_1,x_2,x_3) = \bar{x}_1\bar{x}_3(0) \vee \bar{x}_1x_3(x_2) \vee x_1\bar{x}_3(x_2) \vee x_1x_3(x_2)$

Funkcijski ostanki : $f_0 = 0,\ f_1 = x_2,\ f_2 = x_2,\ f_3 = x_2$

Primerjamo f_0 z $f_1f_2f_3$ $\quad f_1 \supseteq f_0$; $f_2 \supseteq f_0$; $f_3 \supseteq f_0$

Primerjamo f_1 z f_2f_3 $\quad f_1 = f_2$; $f_1 = f_3$

Primerjamo f_2 z f_3 $\quad f_2 = f_3$

Razčlenitev po x_2x_3:

$f(x_1,x_2,x_3) = \bar{x}_2\bar{x}_3f_0 \vee \bar{x}_2x_3f_1 \vee x_2\bar{x}_3f_2 \vee x_2x_3f_3$

$f(x_1,x_2,x_3) = \bar{x}_2\bar{x}_3(0) \vee \bar{x}_2x_3(0) \vee x_2\bar{x}_3(x_1) \vee x_2x_3(1)$

Funkcijski ostanki : $f_0 = 0,\ f_1 = 0,\ f_2 = x_1,\ f_3 = 1$

Primerjamo f_0 z $f_1f_2f_3$ $\quad f_0 = f_1$; $f_2 \supseteq f_0$; $f_3 \supseteq f_0$

Primerjamo f_1 z f_2f_3 $\quad f_2 \supseteq f_1$; $f_3 \supseteq f_1$

Primerjamo f_2 z f_3 $\quad f_3 \supseteq f_2$

Preklopna funkcija je 2 - monotona

Preklopna funkcija je linearno ločljiva, ker zadošča pogoju monotonosti (funkcija je 1- monotona in 2-monotona).

Pogoj neseštevljivosti preklopne funkcije

Preklopna funkcija f(X) je k - seštevljiva, $2 \leq k \leq 2^n/2$, če je možno poiskati niz k vhodnih vektorjev, ki dajo vrednost funkcije 1 in niz k vhodnih vektorjev pri katerih je funkcija 0 tako, da je vektorska vsota prvih enaka vektorski vsoti drugih. Vektorska vsota vhodnih kombinacij pomeni vektor, katerega komponente so algebraična vsota komponent vhodnih vektorjev.

$$\sum_1^k x_T = \sum_1^k x_F$$

T - true (pravilen -1) x_T - oznaka za vektor, ki daje f(x) = 1

F - false (nepravilen -0) x_F - oznaka za vektor, ki daje f(x) = 0

Če ni mogoče najti takšen niz k-pravilnih in k-nepravilnih vektorjev, katerih vsota bi bila enaka, potem je izpolnjen pogoj neseštevljivosti - preklopna funkcija je neseštevljiva.

PRIMER 5.14 Ugotovite ali je podana preklopna funkcija neseštevljiva.

$$f(x_1,x_2,x_3) = x_1x_2 \vee \overline{x}_1x_3 = \underset{(1,1,0)}{x_1x_2\overline{x}_3} \vee \underset{(1,1,1)}{x_1x_2x_3} \vee \underset{(0,0,1)}{\overline{x}_1\overline{x}_2x_3} \vee \underset{(0,1,1)}{\overline{x}_1x_2x_3}$$

vektorji x_T: (1,1,0), (1,1,1), (0,0,1), (0,1,1) - vhodne kombinacije, kjer je f(x) =1

vektorji x_F: (0,0,0), (0,1,0), (1,0,0), (1,0,1) - vhodne kombinacije, kjer je f(x) =0

Pogoj seštevljivosti: k = 2

$\sum x_T = (1,1,0) + (0,0,1) = (1,1,1)$

$\sum x_F = (0,1,0) + (1,0,1) = (1,1,1)$

Vektorska vsota vhodnih kombinacij je enaka, kar pomeni, da je preklopna funkcija 2-seštevljiva.

PRIMER 5.15 Ugotovite ali je podana preklopna funkcija neseštevljiva.

$$f(x_1,x_2,x_3) = x_2 \vee x_1x_3 = \underset{(0,1,0)}{\overline{x}_1x_2\overline{x}_3} \vee \underset{(0,1,1)}{\overline{x}_1x_2x_3} \vee \underset{(1,1,0)}{x_1x_2\overline{x}_3} \vee \underset{(1,1,1)}{x_1x_2x_3} \vee \underset{(1,0,1)}{x_1\overline{x}_2x_3}$$

vektorji x_T so : (0,1,0), (0,1,1), (1,1,0), (1,1,1), (1,0,1)

vektorji x_F so : (0,0,0), (0,0,1), (1,0,0)

Pogoj seštevljivosti: k = 2

$\sum x_T = (0,1,0) + (1,0,1) = (1,1,1)$

$\sum x_F = (0,0,0) + (0,0,1) = (0,0,1) \qquad \sum x_T \neq \sum x_F$

$\sum x_T = (0,1,0) + (1,0,1) = (1,1,1)$

$\sum x_F = (0,0,0) + (1,0,0) = (1,0,0) \qquad \sum x_T \neq \sum x_F$

$\sum x_T = (0,1,0) + (1,0,1) = (1,1,1)$

$\sum x_F = (0,0,1) + (1,0,0) = (1,0,1) \qquad \sum x_T \neq \sum x_F$

Postopek primerjave lahko nadaljujemo še za ostale primere vektorjev x_T, vendar lahko hitro vidimo, da pri nobeni kombinaciji teh vektorjev ni možno tvoriti takšnih vektorskih vsot, da bi bil enak rezultat (primer je za en sam par vektorjev x_T in vse možnosti x_F).

Pogoj seštevljivosti: k = 3

$\sum x_T = (0,1,0) + (1,0,1) + (0,1,1) = (1,2,2)$

$\sum x_F = (0,0,0) + (0,0,1) + (1,0,0) = (1,0,1) \quad \sum x_T \neq \sum x_F$

$\sum x_T = (0,1,0) + (1,0,1) + (1,1,0) = (2,2,1)$

$\sum x_F = (0,0,0) + (0,0,1) + (1,0,0) = (1,0,1) \quad \sum x_T \neq \sum x_F$

Če bi pogledali še za vse ostale primere vektorjev x_T in x_F, ne najdemo nobene enakosti vektorskih vsot za k = 3, zato je preklopna funkcija neseštevljiva.

5.2.2 Realizacija preklopnih funkcij s pragovnimi operatorji

Simetrični polinomi stopnje d so pragovne funkcije, katerih uteži so $a_i = 1$, za $i=1,\dots,n$ in prag P = d.

$f_{(3,4)}(x_1,x_2,x_3,x_4)$ je pragovna funkcija s pragom P = 3

Konjunkcija in disjunkcija sta tudi lahko predstavljeni s pragovnim operatorjem. Za realizacijo poljubne simetrične funkcije nam torej manjka le še negacija in zato lahko definiramo pragovne operatorje skupaj z negacijo kot funkcijsko poln sistem.

$f(x_1,x_2) = x_1 x_2 = f_{(2)}(x_1,x_2)$ konjunkcija je simetrični polinom

Konjunkcija je pragovna funkcija z utežmi $a_i = 1$ za $i = 1,2$ in pragom $P = 2$ za funkcijo z dvema vhodoma.

$f(x_1,x_2) = x_1 \vee x_2 = f_{(1,2)}(x_1,x_2)$ disjunkcija je simetrični polinom

Disjunkcija je pragovna funkcija z utežmi $a_i = 1$ za $i = 1,2$ in pragom $P = 1$ za poljubno število funkcijskih vhodov.

Realizacija simetričnih funkcij s pragovnimi operatorji

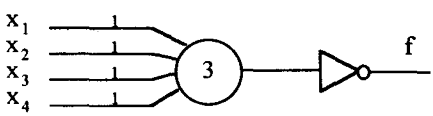
Za realizacijo uporabimo simetrični polinom, kateremu dodamo negacijo, da dobimo funkcijsko poln sistem.

PRIMER 5.16 Realizirajte simetrično preklopno funkcijo z uporabo pragovnih operatorjev tako, da je število le teh minimalno.

$f_{(0,1,2)}(x_1,x_2,x_3,x_4)$

Če simetrično funkcijo negiramo dobimo simetrični polinom.

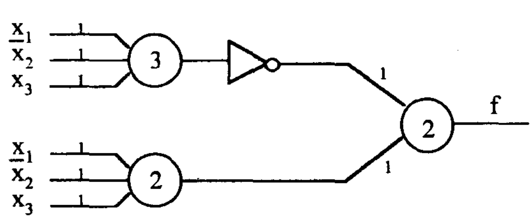
$f = f_{(0,1,2)}(x_1,x_2,x_3,x_4) = \bar{f}_{(3,4)}(x_1,x_2,x_3,x_4)$



PRIMER 5.17 Realizirajte simetrično preklopno funkcijo $f_{(2)}(x_1,\bar{x}_2,x_3)$ z uporabo pragovnih operatorjev tako, da je število le teh minimalno.

Preklopno funkcijo razširimo z operacijama konjunkcije in disjunkcije v več simetričnih polinomov.

$f_{(2)}(x_1,\bar{x}_2,x_3) = f_{(0,1,2)}(x_1,\bar{x}_2,x_3)\ \&\ f_{(2,3)}(x_1,\bar{x}_2,x_3) = \bar{f}_{(3)}(x_1,\bar{x}_2,x_3)\ \&\ f_{(2,3)}(x_1,\bar{x}_2,x_3)$



5.3 Verjetnostne preklopne funkcije

Preklopna funkcija $f(x_1,x_2,...,x_n)$ je verjetnostna, če se spremenljivke x_i pojavljajo z verjetnostmi p $(x_i=1)$ oz. $\overline{p}$ $(x_i=0)$. Verjetnost preklopne funkcije je določena na osnovi vseh možnih dogodkov:

$$P(f) = \sum_{i=0}^{2^n-1} P(m_i)$$

$P(m_i)$ - verjetnost minterma m_i

$P(m_i) = 0$, če minterm m_i ne nastopa v preklopni funkciji

5.3.1 Določanje verjetnostne preklopne funkcije

Pretvorba preklopne funkcije $f(x_1,x_2,...,x_n)$ v verjetnostno funkcijo P(f) je možna v ODNO (ortogonalni disjunktivni normalni obliki) tako, da spremenljivko x_i zamenjamo s p oz. spremenljivko $\overline{x}_i$ z (1-p) in operator konjunkcije s produktom ter operator disjunkcije z vsoto.

ODNO (ortogonalna disjunktivna normalna oblika)

Podana preklopna funkcija je ortogonalna, če v disjunktivni normalni obliki za vsak par konjunkcij velja $q_i q_j = 0$, kjer sta q_i in q_j dva konjunktivna izraza podane preklopne funkcije.

PRIMER 5.18 PDNO (funkcija v PDNO je vedno tudi ODNO):

$$f(x_1,x_2) = \overline{x}_1\overline{x}_2 \vee x_1 x_2$$

$q_i q_j = \overline{x}_1\overline{x}_2 x_1 x_2 = \overline{x}_1 x_1 \overline{x}_2 x_2 = 0$ pogoj ortogonalnosti je izpolnjen

PRIMER 5.19 DNO (funkcija v DNO je lahko ortogonalna, ali pa ne)

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

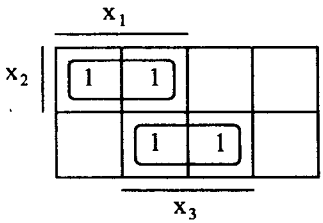
$q_i q_j = x_1 x_2 \overline{x}_1 \overline{x}_2 \overline{x}_3 = 0$ pogoj ortogonalnosti je izpolnjen

$q_i q_j = x_1 x_2 x_3 \neq 0$ pogoj ortogonalnosti ni izpolnjen

Preklopna funkcija v DNO ni ortogonalna, ker ni izpolnjen pogoj ortogonalnosti za vse pare konjunkcij.

PRIMER 5.20 MDNO (minimalna disjunktivna normalna oblika) je ortogonalna takrat, kadar je pri minimizaciji vsaka enica v Veitchevem diagramu upoštevana samo enkrat - imenuje se MODNO.

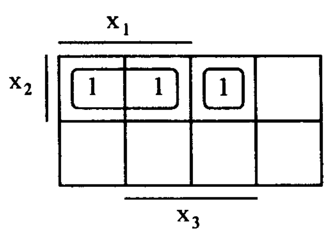
$f(x_1,x_2,x_3) = x_1x_2 \vee \overline{x}_2x_3$



PRIMER 5.21 Izračunajte verjetnost, da podana preklopna funkcija zavzame vrednost 0, če je podana verjetnost $p_i = 0.7$, $i = 1,2,3$, da vhodna spremenljivka x_i zavzame vrednost 1.

$f(x_1,x_2,x_3) = x_1x_2 \vee x_2x_3$

MODNO (minimalna ortogonalna disjunktivna normalna oblika):



MODNO: $f = x_1x_2 \vee \overline{x}_1x_2x_3$

$P(f{=}1) = p_i^2 + p_i^2(1 - p_i) = p_i^2 + p_i^2 - p_i^3 = 2p_i^2 - p_i^3 = 2 \times 0.7^2 - 0.7^3 = 0.98 - 0.343 = 0.637$

$P(f{=}0) = 1 - P(f{=}1) = 1 - 0.637 = 0.363$

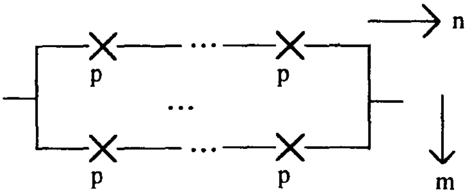
NALOGE:

Izračunajte verjetnost, da preklopna funkcija $f(x_1,x_2,x_3,x_4) = \vee\ (0,2,4,6,12)$ zavzame vrednost 1, če je verjetnost $p_i = 0.6$, $i = 1,2,3,4$, da vhodna spremenljivka x_i zavzame vrednost 1.
R: $P(f{=}1) = 0.2176$

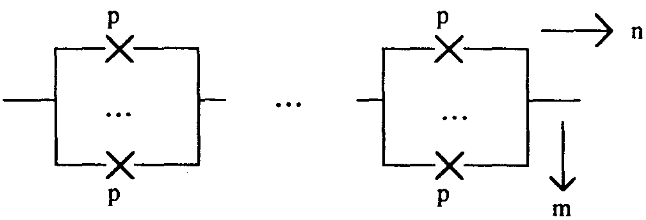
5.3.2 Povečevanje zanesljivosti preklopnih funkcij

Vsak preklopnik v vezju ima podano verjetnost pravilnega delovanja, ki je lahko enaka 1 ali pa manjša. V primeru premajhne zanesljivosti tega elementa jo skušamo povečati tako, da en preklopnik nadomestimo s shemo preklopnikov, katere zanesljivost je večja.

Princip z redundanco

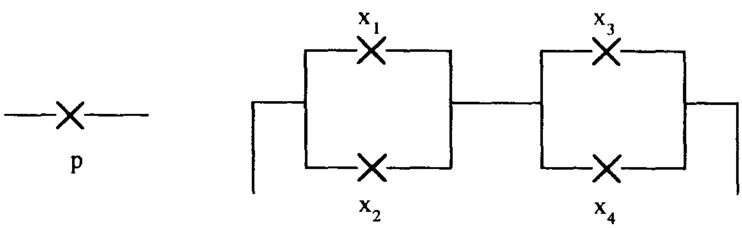


$$P(p) = 1 - (1 - p^n)^m$$



$$P(p) = (1 - (1 - p)^m)^n$$

PRIMER 5.22 Določite verjetnost pravilnega delovanja spodnje sheme, kjer je zanesljivost povečana z uporabo redundatnih elementov.



P(f)

p - verjetnost pravilnega delovanja elementa x
P(f) - verjetnost pravilnega delovanja funkcije f

1. Uporabimo enačbo $P(f) = (1 - (1 - p)^m)^n$ iz redundančne sheme za n=2, m=2

$$P(f) = (1 - (1 - p)^2)^2 = (1 - (1 - 2p + p^2))^2 = (2p - p^2)^2 = 4p^2 - 4p^3 + p^4$$

2. Zapis funkcije za podano shemo in izračun verjetnostne preklopne funkcije:

$f(x_1,x_2,x_3,x_4) = (x_1 \vee x_2).(x_3 \vee x_4) = x_1x_3 \vee x_2x_3 \vee x_1x_4 \vee x_2x_4$

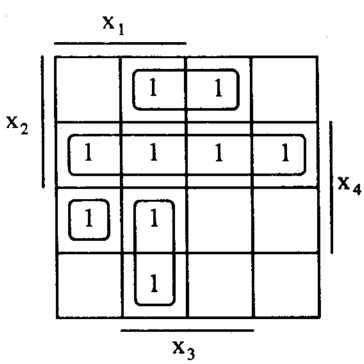
Pogoj ortogonalnosti:

1. DNO → PDNO = ODNO

$$f(x_1,x_2,x_3,x_4) = x_1\overline{x}_2x_3\overline{x}_4 \vee x_1\overline{x}_2x_3x_3 \vee x_1x_2x_3\overline{x}_4 \vee x_1x_2x_3x_4 \vee$$
$$\vee \overline{x}_1x_2x_3\overline{x}_4 \vee \overline{x}_1x_2x_3x_4 \vee x_1\overline{x}_2\overline{x}_3x_4 \vee x_1x_2\overline{x}_3x_4 \vee \overline{x}_1x_2\overline{x}_3x_4$$

$$P(f) = 4p^2(1-p)^2 + 4p^3(1-p) + p^4 = 4p^2(1-2p+p^2) + 4p^3 - 4p^4 + p^4 =$$
$$= 4p^2 - 8p^3(1-p) + 4p^4 + 4p^3 - 4p^4 + p^4 = 4p^2 - 4p^3 + p^4$$

2. DNO → MDNO = ODNO



MODNO: $f(x_1,x_2,x_3,x_4) = x_2x_4 \vee x_2x_3\overline{x}_4 \vee x_1\overline{x}_2x_3 \vee x_1\overline{x}_2\overline{x}_3x_4$

$$P(f) = p^2 + 2p^2(1-p) + p^2(1-p)^2 = p^2 + 2p^2 - 2p^3 + p^2(1-2p+p^2) =$$
$$= p^2 + 2p^2 - 2p^3 + p^2 - 2p^3 + p^4 = 4p^2 - 4p^3 + p^4$$

Pristop z večinskim principom

Za realizacijo je uporabljen večinski operator po von Neumannu, ki daje na izhodu simetrični polinom in se imenuje tudi Lindamanov operator.

Lindamanov ali majoritetni operator:

$$f(x_1,x_2,x_3) = f_{(2,3)}(x_1,x_2,x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3 = x_1 \# x_2 \# x_3$$

Verjetnostna funkcija za Lindamanov operator:

$f_{(2,3)}(x_1,x_2,x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3$

ODNO = PDNO: $f(x_1,x_2,x_3) = x_1x_2\bar{x}_3 \vee x_1x_2x_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3$

Verjetnostna preklopna funkcija: $P(f) = 3p^2(1-p) + p^3 = 3p^2 - 2p^3$

Poljubno preklopno funkcijo je možno realizirati z Lindamanovim operatorjem, konstantama 1,0 in negacijo (#,konst,NEG) z uporabo razčlenjevanja preklopne funkcije. Postopek razčlenjevanja nad funkcijskimi ostanki se ponavlja, dokler njegov rezultat ni ena sama spremenljivka ali konstanta.

Zapis dvovhodne konjunkcije in disjunkcije z Lindamanovim operatorjem:

$x_1x_2 = x_1 \# x_2 \# 0$

$x_1 \vee x_2 = x_1 \# x_2 \# 1$

Razčlenitev po x_1 in zapis z Lindamanovim operatorjem

$f(x_1,x_2,\ldots,x_n) = \bar{x}_1.f_0 \vee x_1.f_1 = (\bar{x}_1 \# f_0 \# 0) \# (x_1 \# f_1 \# 0) \# 1$

PRIMER 5.23 Realizirajte podano preklopno funkcijo z Lindamanovim operatorjem.

$f(x_1,x_2,x_3) = x_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3$

Razčlenitev funkcije po x_1:

$f(x_1,x_2,x_3) = \bar{x}_1.f_0 \vee x_1.f_1 = \bar{x}_1(x_2x_3) \vee x_1(\bar{x}_2x_3)$

Razčlenitev funkcijskih ostankov po x_2:

$f_0 = \bar{x}_2x_3 = \bar{x}_2(x_3) \vee x_2(0)$

$f_1 = x_2x_3 = \bar{x}_2(0) \vee x_2(x_3)$

Funkcija razčlenjena po x_1 se zapiše z Lindamanovim operatorjem, kjer dobimo funkcijska ostanka, ki ju v naslednjem koraku razčlenimo po x_2 spremenljivki in ponovno je rezultat zapisan z Lindamanovim operatorjem. Po drugem koraku so funkcijski ostanki spremenljivka ali konstanta, zato je postopek končan.

$$\begin{aligned} f(x_1,x_2,x_3) &= (x_1 \# (\bar{x}_2x_3) \# 0) \# (\bar{x}_1 \# (x_2x_3) \# 0) \# 1 = \\ &= (x_1 \# ((\bar{x}_2 \# x_3 \# 0) \# (x_2 \# 0 \# 0) \# 1) \# 0) \# \\ &\quad (\bar{x}_1 \# ((\bar{x}_2 \# 0 \# 0) \# (x_2 \# x_3 \# 0) \# 1) \# 0) \# 1 \end{aligned}$$

6 STRUKTURALNA PREKLOPNA VEZJA

Pri kompleksnih logičnih strukturah (vezjih) nam gradniki, kot so vrata OR, AND, NAND, EX-OR, itd. ne zadoščajo več. Uporabili bomo gradnike katerih funkcija je že vnaprej določena s podano strukturo, zato jih lahko imenujemo tudi strukturalni gradniki. Vezja, ki jih dobimo z uporabljenimi gradniki, imenujemo strukturalna preklopna vezja.

6.1 Strukturalna PDNO in pravilnostna tabela

Funkcija strukturalnih gradnikov je opredeljena z matrikami, vektorji in večmestnimi spremenljivkami. V nadaljevanju si oglejmo določanje vektorjev in matrik v pravilnostni tabeli preklopne funkcije ter strukturalen zapis preklopne funkcije.

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

$\dot{x} = |x_1, x_2|$ - vektor vhodnih spremenljivk

$f(\dot{x}) = f(x_1, x_2)$ - izhodna funkcija

$$W = \begin{vmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{vmatrix} \qquad W^T = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{vmatrix}$$ matrika vhodnih kombinacij

$$\dot{f} = \begin{vmatrix} 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}$$ vektor funkcijskih vrednosti

PDNO - popolna disjunktivna normalna oblika v strukturalnem zapisu

$$f = \bigvee_{i=0}^{2^n-1} m_i f_i = \dot{m} \vee \& \dot{f} = (\dot{x} \& \equiv W^T) \vee \& \dot{f}$$ $\dot{m}$ - mintermski vektor

PKNO - popolna konjunktivna normalna oblika v strukturalnem zapisu

$$f = \underset{i=0}{\overset{2^n-1}{\&}} (M_{2^n-1-i} \vee f_i) = \dot{m}_a \& \vee \dot{f} = (\dot{x} \vee \nabla W^T) \& \vee \dot{f}$$ $\dot{m}_a$ - makstermski vektor

PRIMER 6.1 Zapišite preklopno funkcijo iz pravilnostne tabele na strukturalen način z uporabo definicije za PDNO:

$$f = (\dot{x} \& \equiv W^T) \vee \& \dot{f} = (\begin{vmatrix} x_1, x_2 \end{vmatrix} \& \equiv \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{vmatrix}) \vee \& \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \end{vmatrix} = \begin{vmatrix} \overline{x}_1\overline{x}_2, & \overline{x}_1 x_2, & x_1\overline{x}_2, & x_1 x_2 \end{vmatrix} \vee \& \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \end{vmatrix} =$$

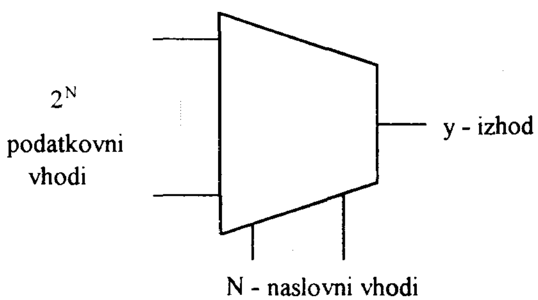
$$= \overline{x}_1\overline{x}_2 0 \vee \overline{x}_1 x_2 1 \vee x_1\overline{x}_2 1 \vee x_1 x_2 0 = \overline{x}_1 x_2 \vee x_1\overline{x}_2$$

6.2 Operatorji srednje integracije (MSI)

Operatorji srednje integracije omogočajo gradnjo kompleksnih preklopnih vezij. Njihova zgradba je podana v strukturalnem zapisu. Pri načrtovanju preklopnih vezij pogosto srečamo gradnike, kot so: multipleksor, demultipleksor, kodirnik, dekodirnik, itd.

6.2.1 Multipleksor (MX) in demultipleksor (DMX)

Multipleksor je MSI logični gradnik (element) z N naslovnimi vhodi in 2^N podatkovnimi vhodi ter enim izhodom. Izhod y je določen s preslikavo podatkovnega vhoda v odvisnosti od izbranega naslova.



Strukturalen zapis izhodne funkcije multipleksorja:

$$y = (\dot{A} \& \equiv D^T) \vee \& \dot{I}$$

$\dot{A}$ - vektor naslovnih spremenljivk
D^T - matrika vhodnih kombinacij za naslovne spremenljivke
$\dot{I}$ - vektor podatkovnih vhodov

Zapis splošne enačbe 1 - naslovnega multipleksorja (1-adr MX)

A_0	y
0	I_0
1	I_1

A_0 - naslovni vhod I_0, I_1 - podatkovna vhoda

izhod: $y = \overline{A}_0 I_0 \vee A_0 I_1$

Zapis splošne enačbe 2 - naslovnega multipleksorja (2-adr MX)

A_1	A_0	y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

A_1, A_0 - naslovna vhoda I_0, I_1, I_2, I_3 - podatkovni vhodi

izhod: $y = \overline{A}_1 \overline{A}_0 I_0 \vee \overline{A}_1 A_0 I_1 \vee A_1 \overline{A}_0 I_2 \vee A_1 A_0 I_3$

Zapis splošne enačbe 3 - naslovnega multipleksorja (3-adr MX)

A_2	A_1	A_0	y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

A_2, A_1, A_0 - naslovni vhodi $I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7$ - podatkovni vhodi

izhod: $y = \overline{A}_2 \overline{A}_1 \overline{A}_0 I_0 \vee \overline{A}_2 \overline{A}_1 A_0 I_1 \vee \overline{A}_2 A_1 \overline{A}_0 I_2 \vee \overline{A}_2 A_1 A_0 I_3 \vee$

$\vee A_2 \overline{A}_1 \overline{A}_0 I_4 \vee A_2 \overline{A}_1 A_0 I_5 \vee A_2 A_1 \overline{A}_0 I_6 \vee A_2 A_1 A_0 I_7$

Postopek realizacije preklopnih funkcij z multipleksorji (MX)

Preklopna funkcija, ki jo želimo realizirati z multipleksorjem je podana z množico spremenljivk $X = (x_1, x_2, \ldots, x_n)$ in izhodom f(X)

Množico spremenljivk bomo razdelili v dva dela: $X = X_{adr} \cup X_{pod}$. V množico naslovnih spremenljivk X_{adr} bomo iz funkcije izbrali toliko spremenljivk, kot imamo naslovnih vhodov v izbranem multipleksorju. Vse preostale spremenljivke v preklopni funkciji postanejo podatkovne, zato jih razvrstimo v množico podatkovnih spremenljivk X_{pod}.

a. naslovne spremenljivke - število naslovnih spremenljivk je določeno z izbranim multipleksorjem :

1 - adr MX ima en naslovni vhod A_0

2 - adr MX ima dva naslovna vhoda (A_1 - MSB bit naslova, A_0 - LSB bit naslova)

3 - adr MX ima tri naslovne vhode (A_2 - MSB bit naslova, A_0 - LSB bit naslova

$X_{adr} = (x_{a1}, x_{a2}, \ldots, x_{aa})$ aa = N - število naslovnih vhodov multipleksorja

b. podatkovne spremenljivke - so vse tiste spremenljivke iz množice X , ki niso izbrane kot naslovne spremenljivke

$$X_{pod} = (x_{d1}, x_{d2}, \ldots, x_{d(n-a)})$$

Za izračun podatkovnih vhodov (0 do $2^N - 1$) uporabimo postopek razčlenjevanja preklopnih funkcij. Podano preklopno funkcijo razčlenimo po tistih spremenljivkah, ki so bile izbrane kot naslovne spremenljivke. Funkcijski ostanki bodo pripeljani na posamezne podatkovne vhode.

Načini realizacije preklopne funkcije z MX-i

1. Trivialna rešitev - vse spremenljivke preklopne funkcije so pripeljane na naslovne vhode multipleksorja, zato sta na podatkovne vhode pripeljani konstanti 0 ali 1.

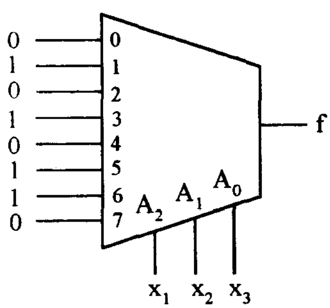
PRIMER 6.2 Preklopno funkcijo $f(x_1, x_2, x_3) = \vee(1,3,5,6)$ realizirajte s 3-naslovnim multipleksorjem. Funkcija (n=3) ≡ 3-adr MX (N =3)

Naslovne spremenljivke so x_1, x_2, x_3 in so pripeljane na naslovne vhode MX-a kot: $A_2 = x_1, A_1 = x_2, A_0 = x_3$. Spremenljivka x_1 predstavlja v spodnji tabeli najbolj pomembno mesto v vhodni kombinaciji funkcije.

Podatkovni vhodi: (0, 1)

Če preklopno funkcijo razčlenimo po treh spremenljivkah dobimo funkcijske ostanke, ki so enaki funkcijskim vrednostim v pravilnostni tabeli. Podatkovni vhodi so določeni s funkcijskimi ostanki, kar pomeni, da so enaki funkcijskim vrednostim v tabeli.

x_1	x_2	x_3	f	podatkovni vhodi
0	0	0	0	$I_0 = 0$
0	0	1	1	$I_1 = 1$
0	1	0	0	$I_2 = 0$
0	1	1	1	$I_3 = 1$
1	0	0	0	$I_4 = 0$
1	0	1	1	$I_5 = 1$
1	1	0	1	$I_6 = 1$
1	1	1	0	$I_7 = 0$



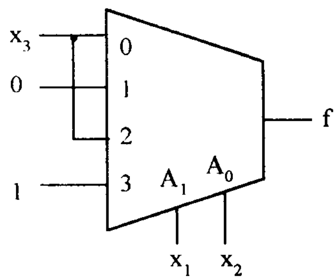
2. Optimalna rešitev - n-1 spremenljivk preklopne funkcije je pripeljanih na naslovne vhode multipleksorja, preostala spremenljivka pa je pripeljana na podatkovne vhode multipleksorja poleg konstant 0 ali 1.

PRIMER 6.3 Preklopno funkcijo $f(x_1,x_2,x_3) = \vee(1,5,6,7)$ realizirajte z 2-naslovnim multipleksorjem. Funkcija (n=3) ≡ 2-adr MX (N =2)
Naslovni spremenljivki sta x_1,x_2 in sta pripeljani na naslovne vhode MX-a kot: $A_1 = x_1, A_0 = x_2$. Spremenljivka x_1 predstavlja v spodnji tabeli najbolj pomembno mesto v vhodni kombinaciji funkcije, če opazujemo samo x_1,x_2.

Podatkovni vhodi: (x_3, 0, 1)

Če preklopno funkcijo razčlenimo po spremenljivkah x_1,x_2 dobimo funkcijske ostanke, ki so odvisni od spremenljivke x_3. Podatkovni vhodi so določeni s funkcijskimi ostanki, ki jih lahko določimo iz pravilnostne tabele. Pri posameznem naslovu za x_1,x_2 opazujemo odvisnost funkcije od podatkovne spremenljivke x_3.

x_1	x_2	x_3	f	podatkovni vhodi
0	0	0	0	
0	0	1	1	$I_0 = x_3$
0	1	0	0	
0	1	1	0	$I_1 = 0$
1	0	0	0	
1	0	1	1	$I_2 = x_3$
1	1	0	1	
1	1	1	1	$I_3 = 1$



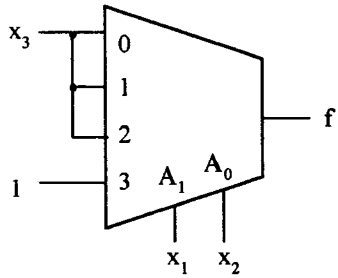
3. Minimalna rešitev - n-2, n-3,.. spremenljivk preklopne funkcije je pripeljanih na naslovne spremenljivke multipleksorja in ena spremenljivka ter konstanti 0,1 na podatkovne vhode. (Takšno rešitev uporabimo takrat, kadar v MDNO preklopna funkcija ni odvisna od vseh spremenljivk).

PRIMER 6.4 Preklopno funkcijo $f(x_1,x_2,x_3,x_4) = x_1x_2 \vee x_3$ (funkcija v MDNO) realizirajte z 2-naslovnim multipleksorjem. Funkcija (n=4) ≡ 2-adr MX (N =2) Naslovni spremenljivki sta x_1,x_2 in sta pripeljani na naslovne vhode MX-a kot: $A_1 = x_1, A_0 = x_2$. Spremenljivka x_1 predstavlja v spodnji tabeli najbolj pomembno mesto v vhodni kombinaciji funkcije, če opazujemo samo x_1,x_2.

Podatkovni vhodi: $(x_3, 0, 1)$

Funkcijo razčlenimo po spremenljivkah x_1,x_2 in dobimo podatkovne vhode:

$I_0 = f(0,0,x_3,x_4) = x_3$
$I_1 = f(0,1,x_3,x_4) = x_3$
$I_2 = f(1,0,x_3,x_4) = x_3$
$I_3 = f(1,1,x_3,x_4) = 1$



4. Kaskadna rešitev - v primeru velikega števila spremenljivk je ponavadi potrebno večje število multipleksorjev za realizacijo preklopne funkcije.
- čista kaskadna realizacija
- kaskada s čim manjšim številom MX-ev
- kaskada s čim manjšim številom nivojev

PRIMER 6.5

Realizirajte preklopno funkcijo $f(x_1,x_2,x_3,x_4,x_5,x_6)=x_1x_2x_3 \vee \bar{x}_1x_2\bar{x}_4\bar{x}_6 \vee x_1\bar{x}_2x_4x_5x_6$ z uporabo 2-naslovnih multipleksorjev (2 - adr MX)

Naslovni spremenljivki: $A_1 = x_1, A_0 = x_2$

Razčlenimo preklopno funkcijo po spremenljivkah x_1,x_2:

$I_0 = f(0,0,x_3,x_4,x_5,x_6) = 0$

$I_1 = f(0,1,x_3,x_4,x_5,x_6) = \bar{x}_4\bar{x}_6 = g$

$I_2 = f(1,0,x_3,x_4,x_5,x_6) = x_4x_5x_6 = h$

$I_3 = f(1,1,x_3,x_4,x_5,x_6) = x_3.$

Razčlenimo funkcijske ostanke, ki niso konstanta ali ena spremenljivka naprej po dveh spremenljivkah, ker jih bomo znova realizirali z 2-adr MX-ji.

$g = \bar{x}_4\bar{x}_6$ naslovni spremenljivki: $A_1 = x_4, A_0 = x_6$

$I_0 = f(0,0) = 1$

$I_1 = f(0,1) = 0$

$I_2 = f(1,0) = 0$

$I_3 = f(1,1) = 0$

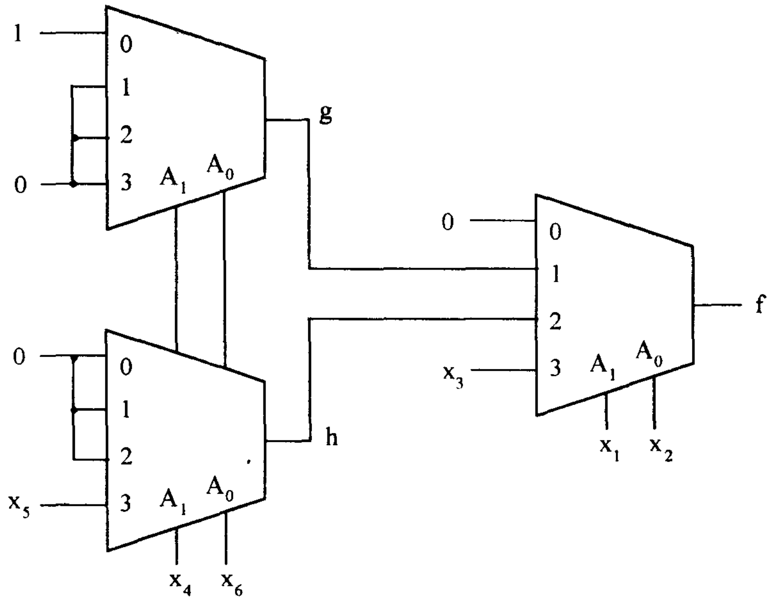
$h = x_4x_5x_6$ naslovni spremenljivki: $A_1 = x_4, A_0 = x_6$

$I_0 = f(0,0,x_5) = 0$

$I_1 = f(0,1,x_5) = 0$

$I_2 = f(1,0,x_5) = 0$

$I_3 = f(1,1,x_5) = x_5$



PRIMER 6.6 $f(x_1,x_2,x_3,x_4,x_5) = x_1\overline{x}_2x_3x_4 \vee \overline{x}_3x_5 \vee x_2x_3\overline{x}_5 \vee \overline{x}_1x_4x_5$

Realizirajte podano preklopno funkcijo z uporabo 2-naslovnih multipleksorjev (2 - adr MX). V funkciji se največkrat pojavljata spremenljivki x_3 in x_5, zato ju vzamemo kot naslovni spremenljivki: $A_1 = x_3, A_0 = x_5$

$I_0 = f(x_1,x_2,0,x_4,0) = 0$
$I_1 = f(x_1,x_2,0,x_4,1) = 1$
$I_2 = f(x_1,x_2,1,x_4,0) = x_1\overline{x}_2x_4 \vee x_2 = h$
$I_3 = f(x_1,x_2,1,x_4,1) = x_1\overline{x}_2x_4 \vee \overline{x}_1x_4 = k$

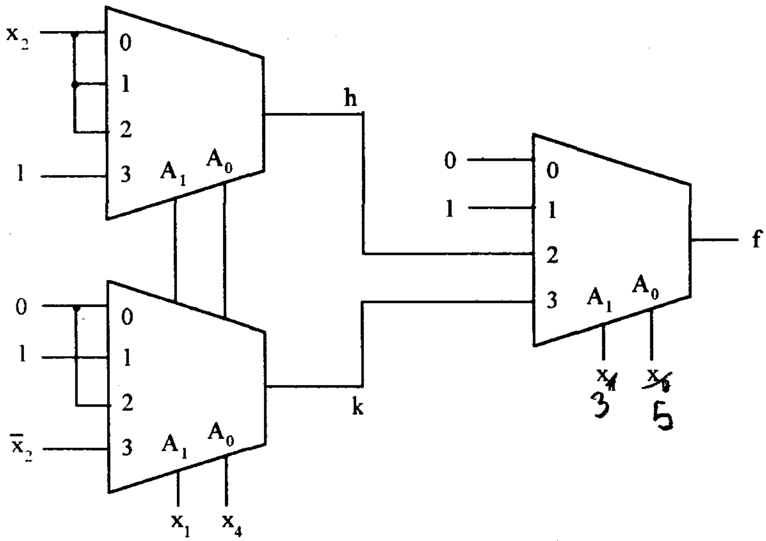
Razčlenimo funkcijska ostanka h in k po spremenljivkah x_1, x_4:

$h = x_1\overline{x}_2x_4 \vee x_2$ naslovni spremenljivki: $A_1 = x_1, A_0 = x_4$

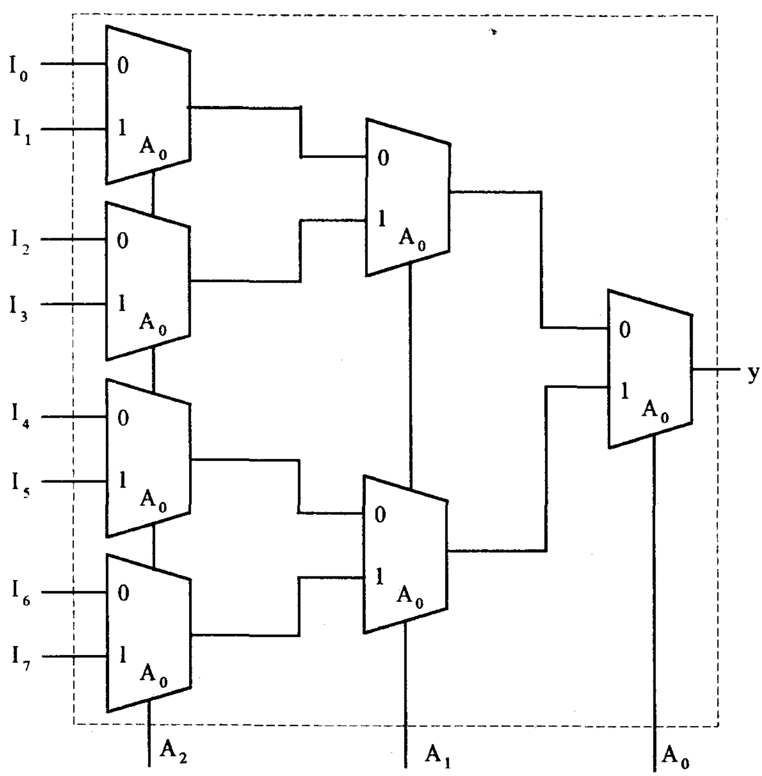
$I_0 = f(0,x_3,0) = x_2$
$I_1 = f(0,x_3,1) = x_2$
$I_2 = f(1,x_3,0) = x_2$
$I_3 = f(1,x_3,1) = 1$

$k = x_1\overline{x}_2x_4 \vee \overline{x}_1x_4$ naslovni spremenljivki: $A_1 = x_1, A_0 = x_4$

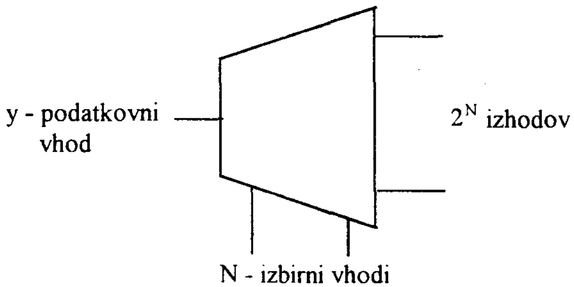
$I_0 = f(0,x_3,0) = 0$
$I_1 = f(0,x_3,1) = 1$
$I_2 = f(1,x_3,0) = 0$
$I_3 = f(1,x_3,1) = \overline{x}_2$



PRIMER 6.7 Realizirajte 3- naslovni multipleksor s kaskadno vezavo 1-naslovnih multipleksorjev



Demultipleksor je gradnik, ki ima obratno funkcijo multipleksorju. Signal na vhodu se prenese na enega od izhodov v odvisnosti od stanja izbirnih vhodov. Imamo torej 2^N izhodov pri N selektivnih (izbirnih) vhodih ter en sam podatkovni vhod.



Strukturalen zapis funkcije demultipleksorja:

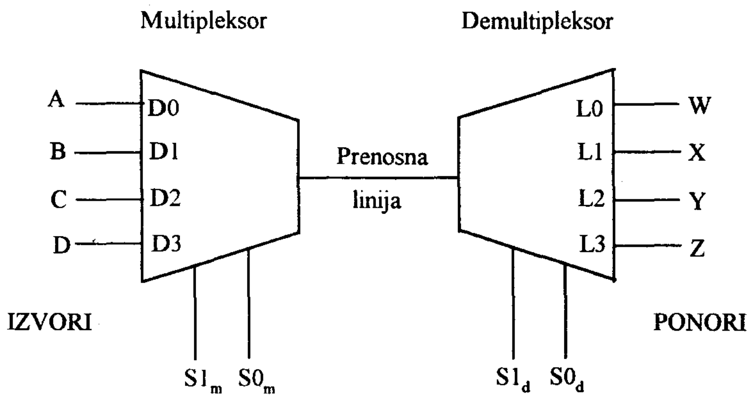
$$\dot{k} = (\dot{x} \& \equiv D^T)^T \vee \& \, y$$

$\dot{k}$ - vektor izhodov

D^T - matrika vhodnih kombinacij za izbirne spremenljivke

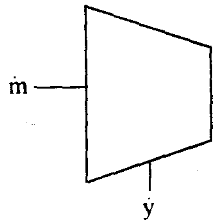
$\dot{x}$ - izbirne spremenljivke

PRIMER 6.8 Z uporabo multipleksorja in demultipleksorja izdelajmo enostaven telekomunikacijski sistem, kjer je lahko poljuben izvor povezan s katerimkoli ponorom preko ene prenosne linije v odvisnosti od izbirnih vhodov.



6.2.2 Kodirnik in dekodirnik

Kodirnik je MSI gradnik, ki daje na izhodu binarno kodo vhoda z logično vrednostjo 1. Za N izhodov kodirnika imamo 2^N vhodov.



Strukturalen zapis funkcije kodirnika: $\dot{y} = \dot{m} \vee \& K$

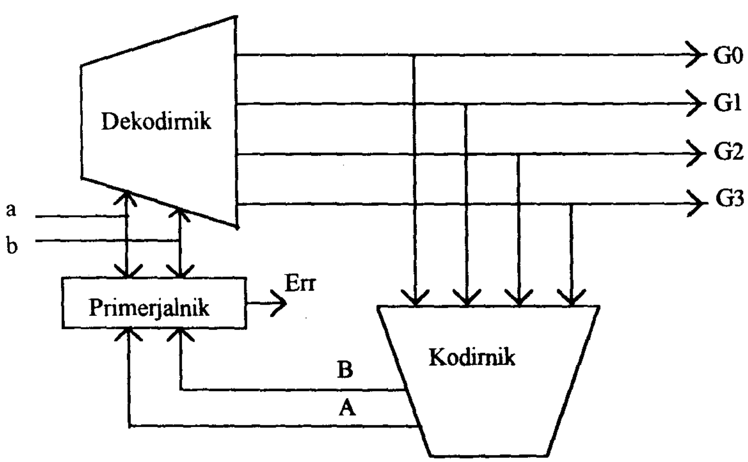
$\dot{m}$ - vektor vhodov
K - matrika binarnih kod
$\dot{y}$ - vektor izhodov

Funkcija kodirnika za generiranje 2-bitne binarne kode:

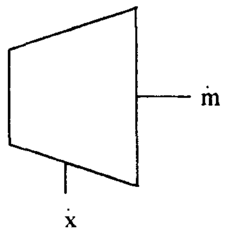
m_0	m_1	m_2	m_3	y_1	y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Na izhodu se pojavi 2-bitna koda signala z logično vrednostjo 1.

PRIMER 6.9: Vezje za testiranje napak dekodirnika. Signal Err na izhodu primerjalnika je po vrednosti 1, če je vhod a ≠ A ali vhod b ≠ B.



Dekodirnik je specialen demultipleksor, ki ima na vhodu vedno vsiljeno vrednost 1. Izbrani izhod ima logično vrednost 1, medtem ko imajo vsi ostali izhodi vrednost 0. Istočasno pa ima dekodirnik obratno funkcijo kot kodirnik.



Strukturalen zapis funkcije dekodirnika: $\dot{m} = \dot{x}\& \equiv D^T$

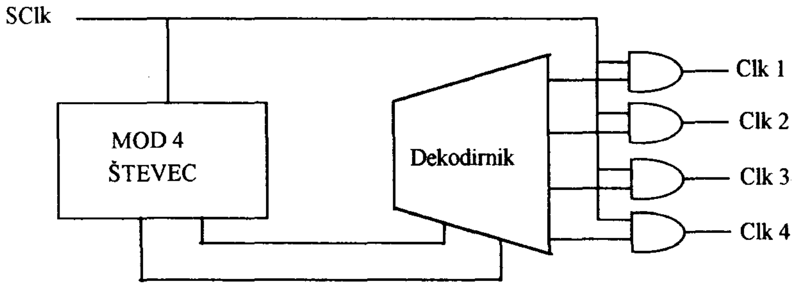
$\dot{m}$ - vektor izhodov
D^T - matrika vhodnih kombinacij za izbirne spremenljivke
$\dot{x}$ - izbirne spremenljivke

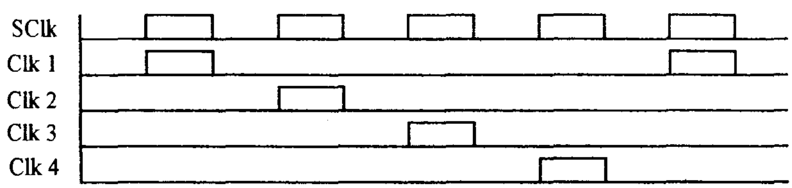
Funkcija dekodirnika z 2 izbirnima vhodoma:

x_1	x_0	m_0	m_1	m_2	m_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Na izhodu se pojavi signal z logično vrednostjo 1, če je bila na vhodu izbrana njegova binarna koda.

PRIMER 6.10 Generator 4-faznega urinega signala s časovnim diagramom.



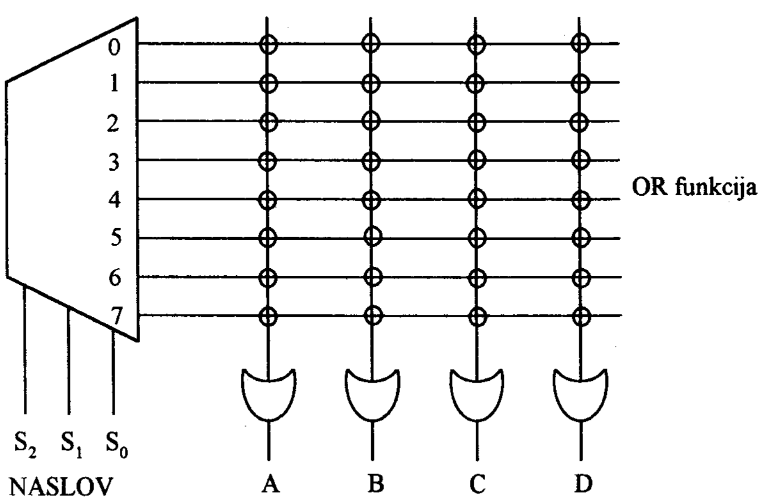


6.3 LSI programabilni logični gradniki

Vsi doslej obravnavani logični gradniki, ki jih uporabljamo za gradnjo preklopnih vezij, imajo točno določeno izhodno funkcijo. Poleg njih pa imamo programabilne logične gradnike, katerim lahko uporabnik sam definira izhode za določeno aplikacijo.
Podrobneje si oglejmo družino programabilnih elementov, kjer programiranje specifičnega problema povzroči fizikalne spremembe znotraj elementa. Vsi omenjeni gradniki so sestavljeni iz dveh tipov vrat (AND in OR). Posamezni tipi se razlikujejo po tem kateri del gradnika je programabilen.
Uporabljene so tri tehnologije programiranja:
- povezave z varovalkami, kjer se ob programiranju odstranijo zahtevane povezave. Takšen element je lahko uporabljen samo za enkratno programiranje.
- elementi brisljivi z ultravioletno svetlobo (UV), kjer lahko po uporabi element zbrišemo in ga znova programiramo. Brisanje elementov z UV traja približno 10 minut.
- električno brisljivi elementi, kjer je čas brisanja skrajšan in je brisanje samo inverzen proces programiranja.

Oglejmo si naslednje tipe programabilnih elementov:

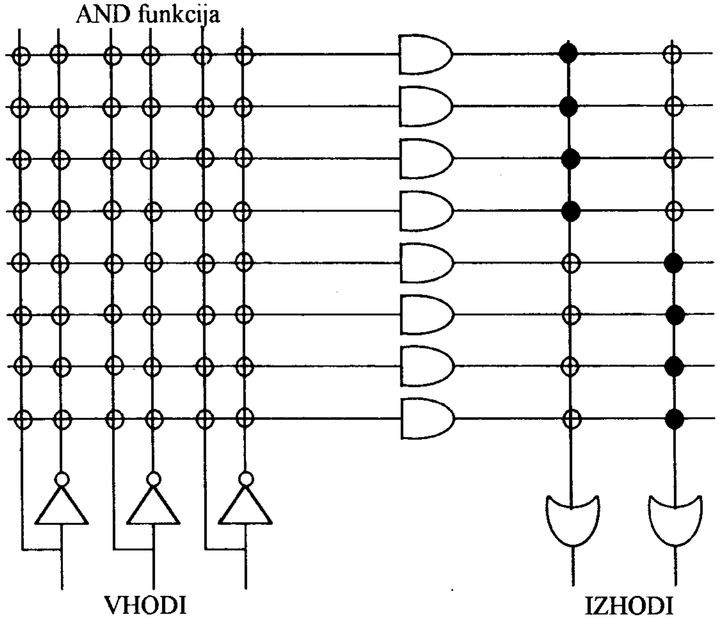
<u>PLE</u>(Programmable Logic Element) - programabilen je OR del gradnika, medtem ko je AND del fiksen. PLE gradnike poznamo pod imeni PROM, EPROM, EEPROM.



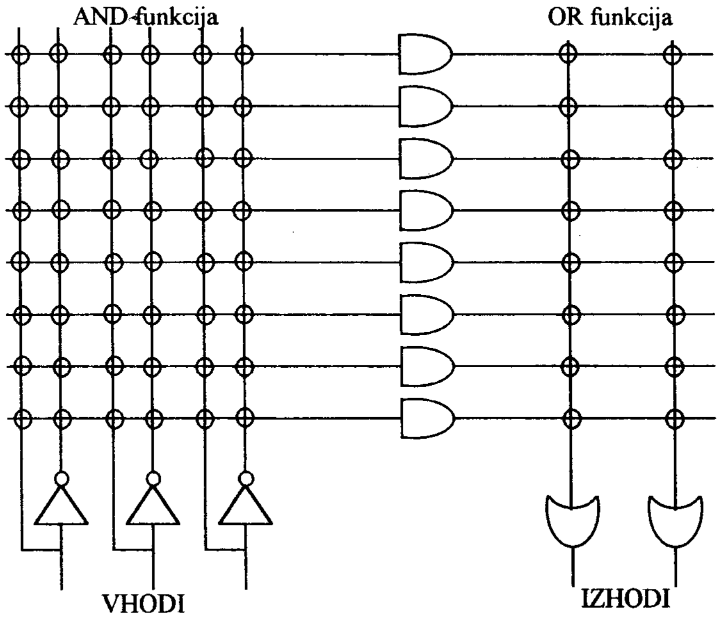
O - predstavlja izbirljivo povezovalno točko

Vsi PROM-i vsebujejo dekoder, katerega izhodi so določeni tako, da je možna povezava vsakega od njih z vsakim vhodom OR vrat.

PAL(Programmable Array Logic) - programabilen je AND del gradnika, medtem ko je OR del fiksen.



PLA(Programmable Logic Array) - programabilna sta oba dela gradnika, tako AND, kot OR del elementa.

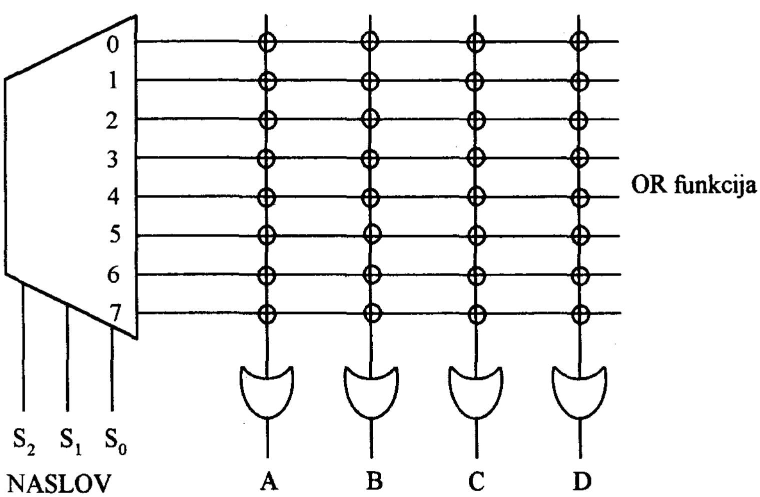


6.3 LSI programabilni logični gradniki

Vsi doslej obravnavani logični gradniki, ki jih uporabljamo za gradnjo preklopnih vezij, imajo točno določeno izhodno funkcijo. Poleg njih pa imamo programabilne logične gradnike, katerim lahko uporabnik sam definira izhode za določeno aplikacijo.
Podrobneje si oglejmo družino programabilnih elementov, kjer programiranje specifičnega problema povzroči fizikalne spremembe znotraj elementa. Vsi omenjeni gradniki so sestavljeni iz dveh tipov vrat (AND in OR). Posamezni tipi se razlikujejo po tem kateri del gradnika je programabilen.
Uporabljene so tri tehnologije programiranja:
- povezave z varovalkami, kjer se ob programiranju odstranijo zahtevane povezave. Takšen element je lahko uporabljen samo za enkratno programiranje.
- elementi brisljivi z ultravioletno svetlobo (UV), kjer lahko po uporabi element zbrišemo in ga znova programiramo. Brisanje elementov z UV traja približno 10 minut.
- električno brisljivi elementi, kjer je čas brisanja skrajšan in je brisanje samo inverzen proces programiranja.

Oglejmo si naslednje tipe programabilnih elementov:

PLE(Programmable Logic Element) - programabilen je OR del gradnika, medtem ko je AND del fiksen. PLE gradnike poznamo pod imeni PROM, EPROM, EEPROM.

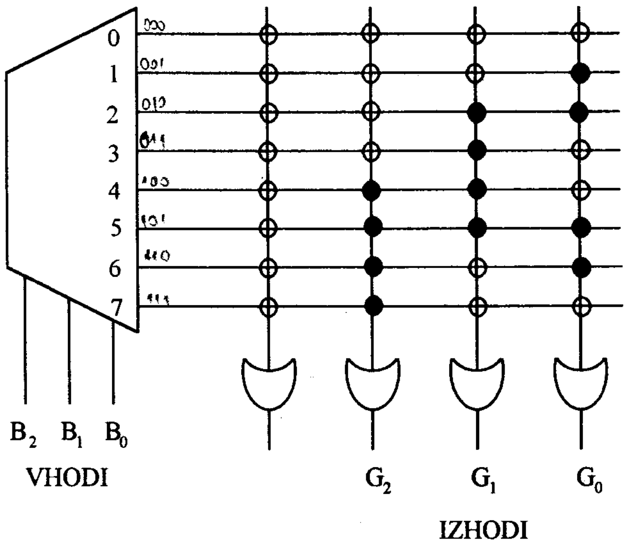


O - predstavlja izbirljivo povezovalno točko

Vsi PROM-i vsebujejo dekoder, katerega izhodi so določeni tako, da je možna povezava vsakega od njih z vsakim vhodom OR vrat.

PRIMER 6.9 Struktura PROM-a je prikazana za pretvorbo 3-bitne binarne kode v 3-bitno Gray-evo kodo.

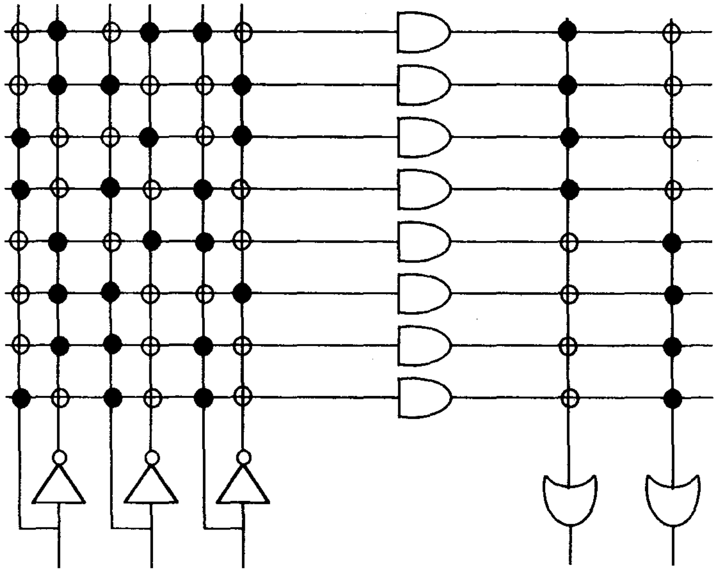
B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0



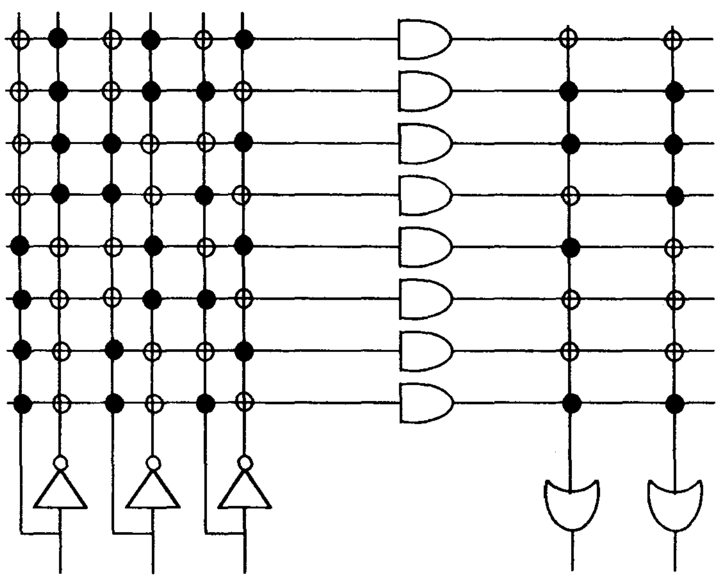
● - povezano　　　○ - nepovezano

PRIMER 6.10 Struktura PAL-a je prikazana za 1-bitni odštevalnik na osnovi pravilnostne tabele.

x	y	B_i	d	B_{i+1}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



PRIMER 6.11 Struktura PLA je prikazana za 1-bitni odštevalnik.



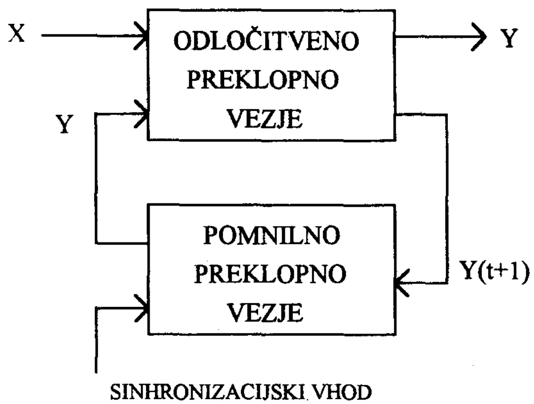
7 SEKVENČNA PREKLOPNA VEZJA

V digitalnih sistemih se pogosto srečamo s problemom shranjevanja ukazov ali podatkov, ki jih bomo uporabili za kasnejše procesiranje. Vezje z dodatno funkcijo pomnjenja imenujemo sekvenčno preklopno vezje. Za sekvenčna preklopna vezja je značilno, da je naboru neodvisnih vhodnih spremenljivk dodana časovna spremenljivka.

$f(x_1,x_2,...,x_n)$ - odločitvena preklopna funkcija

$f(x_1,x_2,...,x_n,t)$ - sekvenčna preklopna funkcija, kjer je t zvezna spremenljivka

Sekvenčno preklopno vezje sestavljajo:
- odločitveno preklopno vezje, ki določa izhodne funkcije
- pomnilno vezje, kjer imamo shranjene podatke (informacije)



Pomnilni del je sestavljen iz večjega števila celic oz. v skrajnem primeru je lahko ena sama pomnilna celica .

Sekvenčna preklopna vezja lahko delimo na:
- sinhronska, kjer se izhod spreminja z določenim časovnim intervalom,
- asinhronska, kjer se izhod spreminja z vhodnimi signali

V nadaljevanju se bomo spoznavali s sekvenčnimi preklopnimi vezji.

7.1 Pomnilne celice

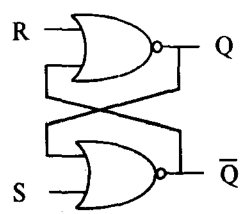
Pomnilna celica je element, kjer je shranjen en bit informacije. Vsaka celica ima v splošnem izhod Q in njegovo negacijo $\overline{Q}(t)$, tako imamo vedno prisotni obe vrednosti pomnjene informacije. Izhod se spreminja v odvisnosti od vhodov. Pri standardnih pomnilnih celicah imamo naslednje krmilne značilnosti:

D - zakasnitev (delay)
T - proženje celice (triggering)
R - pogojno brisanje celice (resetting)
S - pogojno postavljanje celice (setting)
K - brezpogojno brisanje celice
J - brezpogojno postavljanje celice

7.1.1 Osnovne strukture pomnilnih celic

Osnovna struktura vsake pomnilne celice je v povratni povezavi dveh enakih operatorjev (NAND ali NOR). Poglejmo si osnovno strukturo RS pomnilne celice, ki jo imenujemo tudi FF - Flip-Flop.

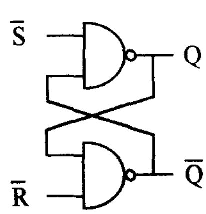
1. Realizacija z NOR operatorji



Vezje ima dva vhoda R in S, ki se nanašata na brisanje (R) in postavljanje (S).

R	S	Q(t+1)
0	0	Q(t)
0	1	1
1	0	0
1	1	prepovedana kombinacija

2. Realizacija z NAND operatorji



Vhoda sta ravno negirana v primerjavi s funkcijo pri vezavi NOR operatorjev.

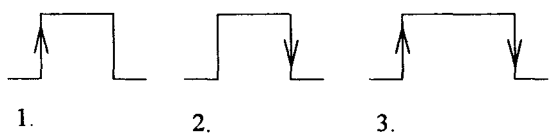
$\overline{R}$	$\overline{S}$	Q(t+1)
0	0	prepovedana kombinacija
0	1	0
1	0	1
1	1	Q(t)

7.1.2 Sinhronske pomnilne celice

V praksi so pomnilne celice definirane z urinim impulzom (Clk), zato jih imenujemo sinhronske in lahko spremenijo svoje stanje (izhod) takrat, kadar je na vhodu prisoten urin impulz. Urin impulz je ozek signal pravokotne oblike, kar pomeni, da ima definirani dve fronti. Prednja fronta je določena ob prehodu signala iz 0 v 1 in zadnja fronta ob prehodu signala iz 1 v 0.

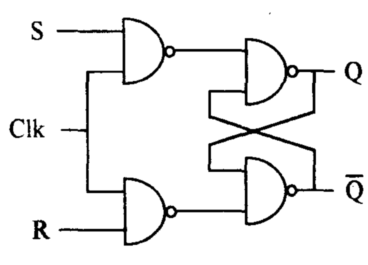
Tipi sinhronizacije:

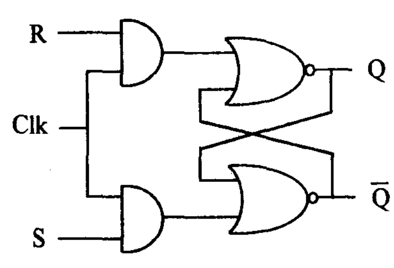
1- prednja fronta
2- zadnja fronta
3- Master-Slave sinhronizacija



PRIMER 7.1 Sinhronska RS pomnilna celica, ki je izpeljana iz osnovnih struktur pomnilnih celic

Iz osnovne strukture pomnilnih celic dobimo sinhronsko pomnilno celico tako, da dodamo na vhode dodaten nivo operatorjev. Urin signal (Clk) je potem pripeljan na vhod vsakega uperatorja, medtem ko je drugi vhod definiran s R in S. Pri povratni vezavi NAND operatorjev smo na predhodni nivo enak operator in odpravili negaciji R, S vhodov.





7.1.3 Določitev izhodnih funkcij za pomnilne celice

Tipi sinhronskih pomnilnih celic:

JK sinhronska pomnilna celica
RS sinhronska pomnilna celica
D sinhronska pomnilna celica
T sinhronska pomnilna celica

Izhodna funkcija pomnilne celice določa spremembe izhoda v odvisnosti od vhodov pomnilne celice in urinega signala.

JK - pomnilna celica

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q}(t)$

J

K

1			
1	1	1	

Q

$Q(t+1) = \overline{K}Q(t) \vee J\overline{Q}(t)$

RS - pomnilna celica

R	S	Q(t+1)
0	0	Q(t)
0	1	1
1	0	0
1	1	×

R

S

x	x	1	1
		1	

Q

$Q(t+1) = S \vee \overline{R}Q(t)$; pogoj $RS = 0$

D - pomnilna celica

D	Q(t+1)
0	0
1	1

$Q(t+1) = D$

T - pomnilna celica

T	Q(t+1)
0	Q(t)
1	$\overline{Q}(t)$

$Q(t+1) = \overline{T}Q(t) \vee T\overline{Q}(t)$

7.1.4 Določanje vhodnih funkcij za pomnilne celice (D, T, RS, JK)

Pri načrtovanju sekvenčnih preklopnih vezij pridemo do problema določanja vhodnih funkcij za pomnilne celice. Vhodne funkcije pomnilne celice so povezane z vezjem, ki ga razvijamo. Za osnovo si poglejmo izračun vhodnih funkcij z uporabo splošne pomnilne enačbe.

Splošna pomnilna enačba: $Q(t+1) = g_1Q(t) \vee g_2\overline{Q}(t)$

V binarno aplikacijsko tabelo zapišemo funkcijo splošne pomnilne enačbe, kjer vidimo da je izhod Q(t+1) odvisen od neodvisnih vhodnih spremenljivk g1, g2 in prejšnjega izhoda Q(t). Če želimo sedaj splošno pomnilno enačbo realizirati z eno od pomnilnih celic moramo izračunati, funkcijske vrednosti vhoda tako, da je zagotovljen izhod Q(t+1) kot je podano v tabeli.

Oglejmo si pomožne tabele prehodov za določitev vhodnih funkcij. V tabelah vidimo določitev vhodov pomnilne celice pri prehodih izhodov iz Q(t) v Q(t+1):

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q}(t)$

Q(t)→Q(t+1)	J	K
0 →0	0	×
0 →1	1	×
1 →0	×	1
1 →1	×	0

R	S	Q(t+1)
0	0	Q(t)
0	1	1
1	0	0
1	1	× - pr.k

Q(t)→Q(t+1)	R	S
0 →0	×	0
0 →1	0	1
1 →0	1	0
1 →1	0	×

T	Q(t+1)
0	Q(t)
1	$\overline{Q}(t)$

Q(t)→Q(t+1)	T
0 → 0	0
0 → 1	1
1 → 0	1
1 → 1	0

D	Q(t+1)
0	0
1	1

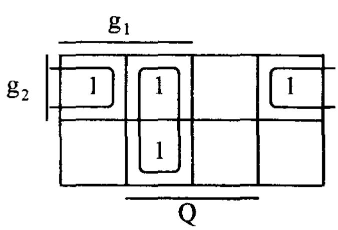
Q(t)→Q(t+1)	D
0 → 0	0
0 → 1	1
1 → 0	0
1 → 1	1

Sedaj lahko definiramo vhodne funkcije za splošno pomnilno enačbo:

g_1	g_2	Q(t)	Q(t+1)	D	T	R	S	J	K
0	0	0	0	0	0	×	0	0	×
0	0	1	0	0	1	1	0	×	1
0	1	0	1	1	1	0	1	1	×
0	1	1	0	0	1	1	0	×	1
1	0	0	0	0	0	×	0	0	×
1	0	1	1	1	0	0	×	×	0
1	1	0	1	1	1	0	1	1	×
1	1	1	1	1	0	0	×	×	0

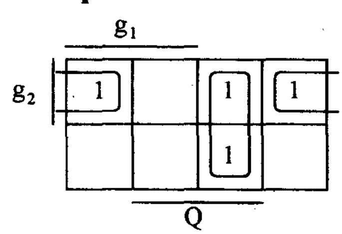
Vhodne funkcije za pomnilne celice so:

D



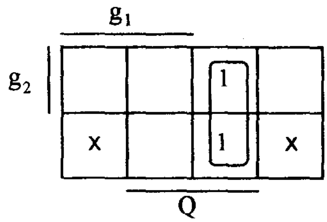
$D = g_1 Q(t) \vee g_2 \overline{Q}(t)$

T

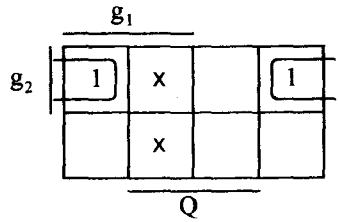


$T = \overline{g}_1 Q(t) \vee g_2 \overline{Q}(t)$

RS

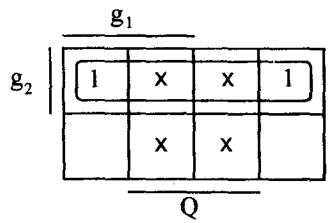


$R = \overline{g}_1 Q(t)$

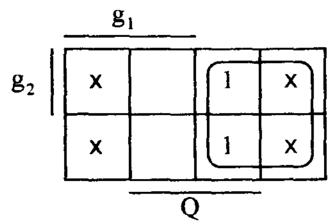


$S = g_2 \overline{Q}(t)$

JK



$J = g_2$



$K = \overline{g}_1$

PRIMER 7.2 Realizirajte sinhronsko D pomnilno celico z uporabo NAND (Shefferjevih) operatorjev.

D pomnilna celica : $Q(t+1) = D$

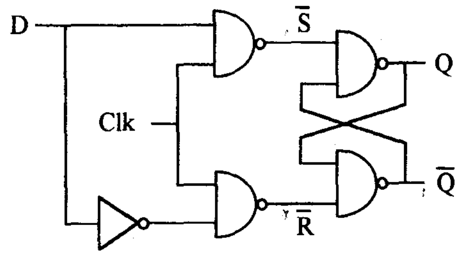
Za osnovo vzamemo sinhronsko RS pomnilno celico z NAND operatorji

D	Q(t)	Q(t+1)	$\overline{R}$	$\overline{S}$
0	0	0	0	1
0	1	0	0	1
1	0	1	1	0
1	1	1	1	0

$\overline{R} = D = \overline{\overline{D}} \vee \overline{Clk} = (\overline{D} \uparrow Clk)$

$\overline{S} = \overline{D} = \overline{D} \vee \overline{Clk} = (D \uparrow Clk)$

Na vhoda za R in S smo dodali sinhronski vhod CLK tako, da imamo dodaten nivo NAND operatorjev.

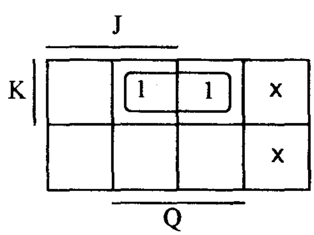


PRIMER 7.3 Realizirajte JK - pomnilno celico z uporabo sinhronske RS pomnilne celice in 1- naslovnih multipleksorjev.

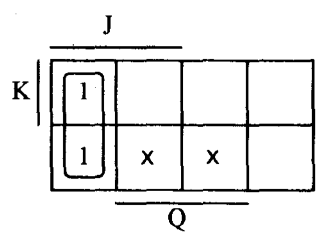
Zapišemo binarno aplikacijsko tabelo za celico, ki jo želimo realizirati. Vhodne spremenljivke so torej J, K in Q(t). Izračunati moramo vhodne funkcije v RS pomnilno celico, ki jo uporabimo v realizaciji.

J	K	Q(t)	Q(t+1)	R	S
0	0	0	0	×	0
0	0	1	1	0	×
0	1	0	0	×	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	0	×
1	1	0	1	0	1
1	1	1	0	1	0

Minimizacija vhodnih funkcij za RS - pomnilno celico

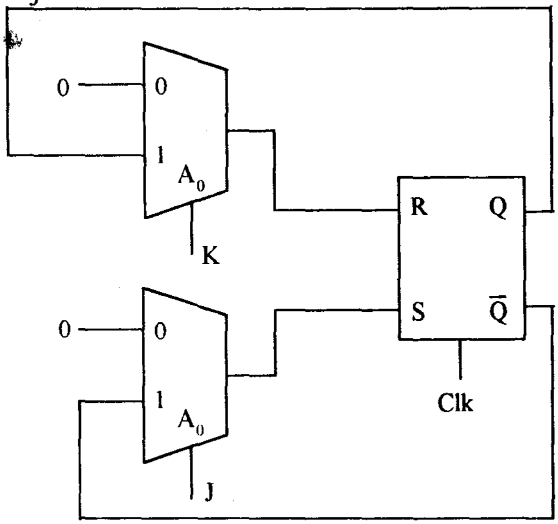


$R = KQ(t)$



$S = J\overline{Q}(t)$

Realizacija:



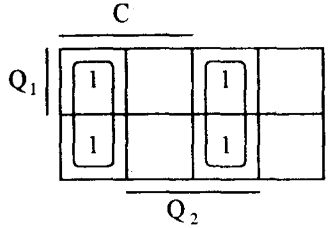
7.2 Realizacija sekvenčnega vezja

PRIMER 7.4
Realizacija sinhronskega up/down števca po modulu 4 z uporabo T - pomnilnih celic in Shefferjevih (NAND) operatorjev.

Izberemo si vhodno spremenljivko C, ki določa funkcijo štetja.

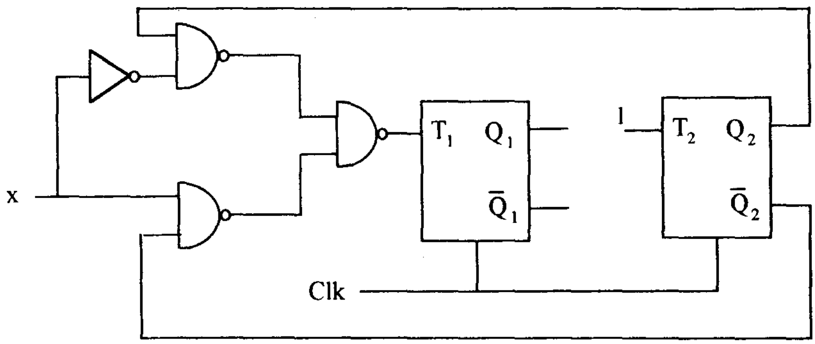
Zapišimo binarno aplikacijsko tabelo za izračun vhodnih funkcij v pomnilne celice T, s katerimi bomo realizirali števec. Izhodi števca so časovne spremenljivke.

C	$Q_1(t)$	$Q_2(t)$	$Q_1(t+1)$	$Q_2(t+1)$	T_1	T_2
0	0	0	0	1	0	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	0	0	1	1
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	1	0	0	1	1	1
1	1	1	1	0	0	1



$T_1 = \overline{C}Q_2(t) \vee C\overline{Q}_2(t)$

$T_2 = 1$



PRIMER 7.5

Realizacija 2- bitnega registra $Y = (y_1, y_2)$, ki ima naslednje funkcije:

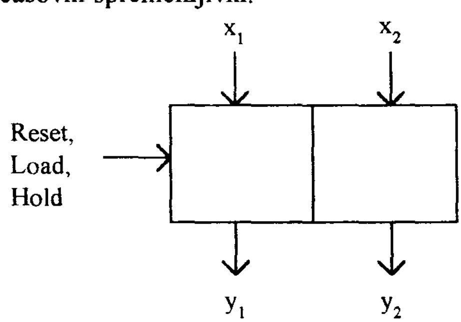
Reset - brisanje registra

Load - vpis 2-bitnega podatka $X = (x_1, x_2)$

Hold - ohranjanje stare vrednosti registra

Realizacija registra s sinhronskima D- pomnilnima celicama in multipleksorji.

Narišimo enostavno blok shemo vezja, ki ga želimo realizirati. Na shemi imamo tri krmilne funkcije (Reset, Load, Hold), vhodni spremenljivki x_1, x_2 in izhoda vezja y_1, y_2, ki sta časovni spremenljivki.



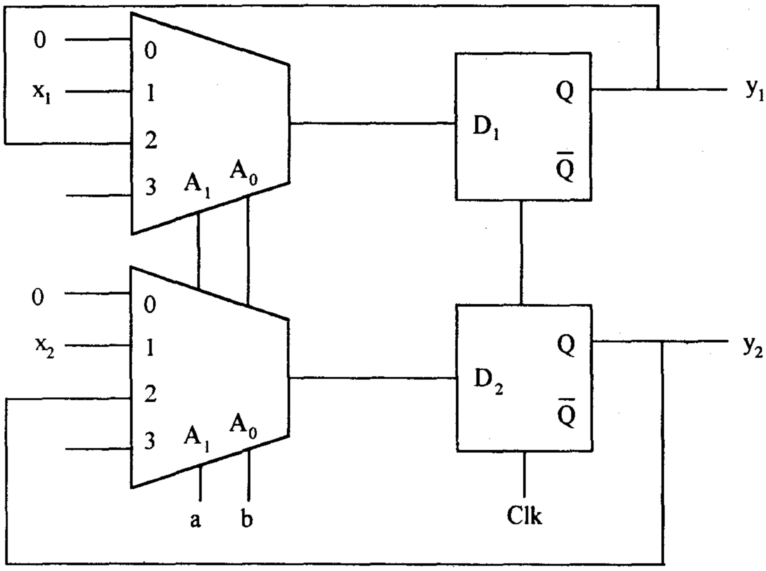
Krmilne funkcije bomo kodirali, zato da zagotovimo minimalno število potrebnih vhodov v vezje. Vidimo, da nam za določanje operacij zadoščata dve krmilni spremenljivki a, b.

a	b	
0	0	Reset
0	1	Load
1	0	Hold
1	1	×

Zapišimo binarno aplikacijsko tabelo za izračun vhodnih funkcij v pomnilne celice D, s katerimi bomo realizirali register.

a	b	$y_1(t)$	$y_2(t)$	$y_1(t+1)$	$y_2(t+1)$	D_1	D_2
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	x_1	x_2	x_1	x_2
0	1	0	1	x_1	x_2	x_1	x_2
0	1	1	0	x_1	x_2	x_1	x_2
0	1	1	1	x_1	x_2	x_1	x_2
1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	1	1	1
1	1	0	0	×	×	×	×
1	1	0	1	×	×	×	×
1	1	1	0	×	×	×	×
1	1	1	1	×	×	×	×

Vhodni funkciji za D pomnilni celici sta enaki izhodoma y v času t+1. Realizacijo vhodnih funkcij z multipleksorji poznamo, zato ne bomo posebej pisali vseh funkcijskih ostankov, če funkcijo razčlenimo po spremenljivkah a in b, ampak jih iz tabele direktno upoštevamo v realizaciji. Podatkovni vhod 3 pri multipleksorjih nima definirane nobene vrednosti, ker lahko zavzame karkoli (×). Realizacija je prikazana z logično shemo.


0
x_1
0
1
2
3
A_1
A_0
D_1
Q
$\overline{Q}$
y_1
0
x_2
0
1
2
3
A_1
A_0
D_2
Q
$\overline{Q}$
y_2
a
b
Clk

8 AVTOMATI

Funkcijo, katere vrednost je odvisna od prejšnjih vrednosti vhodne spremenljivke, kakor tudi od trenutnih vrednosti, imenujemo sekvenčna preklopna funkcija. Pri obravnavanju sekvenčnih preklopnih funkcij govorimo torej o preslikavi med vhodno sekvenco in izhodno sekvenco. Za opis teh funkcij definiramo sekvenčni stroj (avtomat), kjer je vpliv vseh prejšnjih vhodov na izhod predstavljen s stanjem avtomata. Avtomat bomo v nadaljevanju opisovali z množico vhodov X, z množico stanj S in množico izhodov Y.

$X = (x_1, x_2, \ldots, x_n)$ - množica vhodov (vhodnih črk)

$Y = (y_1, y_2, \ldots, y_m)$ - množica izhodov (izhodnih črk)

$S = (S_1, S_2, \ldots, S_p)$ - množica stanj avtomata

Izhod avtomata je v vsakem času odvisen od stanja in vhoda, ki istočasno določata naslednje stanje avtomata. Obravnavali bomo avtomate s končnim številom vhodov, izhodov in stanj, zato jih imenujemo tudi **končni avtomati**.

Za avtomate je značilno, da lahko menjajo stanje ob spremembi vhoda, ki je lahko naključen in v tem primeru imamo **asinhronske avtomate.** Bolj pogosti so **sinhronski avtomati,** ki spreminjajo stanje periodično ob prihodu urinega signala. V nadaljevanju bomo obravnavali sinhronske končne avtomate.

Operacije avtomatov bomo v nadaljevanju opisovali s tabelo stanj. V tabeli vrstice predstavljajo stanja avtomata, kolone pa se nanašajo na vhode. V vrstici stanja S_i je pri vhodu x_j vpisano naslednje stanje in izhod avtomata ob upoštevanju vhoda v stanju S_i.

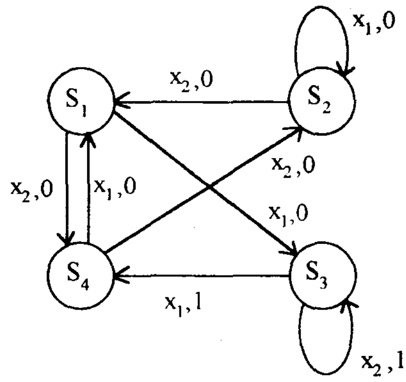
Tabela prehajanja stanj avtomata:

	x_1	x_2
S_1	S_3, 0	S_4, 0
S_2	S_2, 0	S_1, 0
S_3	S_4, 1	S_3, 1
S_4	S_1, 0	S_2, 0

naslednje stanje avtomata, izhod

trenutno stanje avtomata

Drugi način predstavitve operacij avtomata je graf, ki ga imenujemo diagram prehajanja stanj. Vozlišča v grafu predstavljajo stanja avtomata, medtem ko se povezave med stanji nanašajo na prehode stanj. Ob prehodu je podan vhod, ki povzroči spremembo stanja in določi ustrezen izhod.



Tabelo prehajanja stanj ali diagram prehajanja stanj uporabimo za določanje izhodne sekvence, ki je generirana s sekvenco vhodov in začetnim stanjem.

Določimo izhodno sekvenco, če imamo podano vhodno sekvenco $x_1x_2x_1x_2$, ki je pripeljana na začetno stanje S_1. Izhodna sekvenca, ki jo dobimo iz zgornje tabele prehajanja stanj je 0110. Poglejmo si kako smo prišli do dobljene izhodne sekvence. Če je avtomat v začetnem stanju S_1 in imamo vhod x_1 iz podane sekvence, potem iz tabele prehajanja stanj vidimo, da bo izhod 0 in naslednje stanje bo S_3. V tem stanju imamo vhod x_2 iz sekvence, ki definira izhod 1 in avtomat ostane v stanju S_3. Tretji vhod x_1 sedaj določa izhod 1 in prehod avtomata v stanje S_4. Zadnji vhod x_2 pri stanju S_4 določa izhod 0 in prehod v stanje S_2, ki je končno stanje avtomata.

V tabeli prehajanja stanj lahko najdemo tudi nedefinirane (redundančne) prehode stanj. To predstavlja pogoje, kjer naslednje stanje in/ali izhod ni potrebno določiti za prehod iz stanja S_i, vhod x_j, ker se vhod ne more pojaviti, ali pa ni dovoljen kadar je avtomat v stanju S_i. Lahko imamo tudi takšne pogoje, kjer je prehod nepomemben, če se pojavi vhod x_j v stanju S_i.

	x_1	x_2	x_3
S_1	S_1, 0	S_4, 0	S_2, 0
S_2	S_4, 1	-	S_3, 0
S_3	S_5, 1	-	S_3, 1
S_4	S_5 ,1	S_4, 1	S_3, 1
S_5	S_1 ,0	S_1, 0	S_3, 1

V tabeli prehajanja stanj imamo nedefiniran prehod za vhodno sekvenco x_3x_2. S pojavom vhodne črke x_3 preide avtomat vedno v stanje S_2 ali S_3 in pri pojavu vhodne črke x_2 je prehod avtomata nedoločen, zato je ta sekvenca na vhodu prepovedana.

Poznamo dva tipa sinhronskih avtomatov, ki se razlikujeta po tem kako določimo izhod:

Mealyjev avtomat - izhod je odvisen od vhoda in od stanja avtomata
Mooreov avtomat - izhod je odvisen samo od stanja avtomata

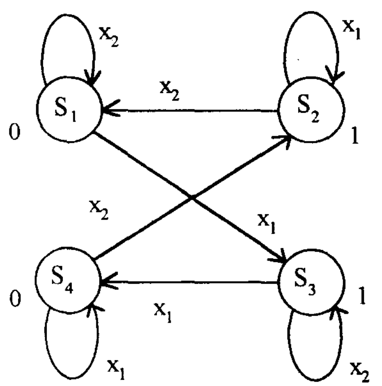
Mooreov avtomat | Mealyjev avtomat

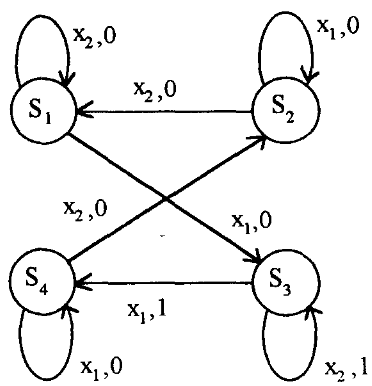
Tabela prehajanja stanj

	x_1	x_2	y
S_1	S_3	S_1	0
S_2	S_2	S_1	1
S_3	S_4	S_3	1
S_4	S_4	S_2	0

	x_1	x_2
S_1	S_3, 0	S_1, 0
S_2	S_2, 0	S_1, 0
S_3	S_4, 1	S_3, 1
S_4	S_4, 0	S_2, 0

Diagram prehajanja stanj





8.1 Ekvivalenca končnih avtomatov

Najpogosteje imamo podane postopke v zvezi s končnimi avtomati za Mooreov avtomat, Do ostalih tipov lahko pridemo z ekvivalenco avtomatov.
Dva končna avtomata sta ekvivalentna, če imata pri enakih sekvencah vhodov enako sekvenco izhodov. Pri tem pa izvzamemo prvo črko na izhodu Mooreovega avtomata in ne sodi v opazovano izhodno sekvenco.

8.1.1 Pretvorba Mealy (ME) → Moore (MO)

Razpredelnica stanj za Mooreov avtomat je določena na osnovi povezave stanj z vhodno črko. Število stanj je odvisno od tega koliko različnih izhodnih črk definira prehod v posamezno stanje Mealyjevega avtomata.

PRIMER 8.1 Pretvorba Mealyjevega avtomata v Mooreov avtomat.

ME	x_1	x_2
S_1	S_2, 1	S_3, 0
S_2	S_5, 1	S_3, 1
S_3	S_1, 0	S_5, 0
S_4	S_2, 1	S_4, 1
S_5	S_3, 0	S_2, 1

V tabeli opazujemo naslednje stanje avtomata s pripadajočim izhodom. Vsako stanje S_i Mealyjevega avtomata lahko ima pripisanih več izhodnih črk (S_i,0 ali S_i,1), kar pomeni, da bomo dobili dve novi stanji Mooreovega avtomata. Na takšen način določimo vsa stanja Mooreovega avtomata. Če znotraj tabele ne najdemo vseh stanj, je takšno stanje potrebno dodati na koncu tabele kot stanje z nedoločenim izhodom S_i,-. Indeksi Mooreovih stanj so dvojni, kjer je prva cifra indeks Mealyjevega stanja in druga cifra indeks izhoda.

MO	x_1	x_2	y
S_{10}	S_{21}	S_{30}	0
S_{21}	S_{51}	S_{31}	1
S_{30}	S_{10}	S_{50}	0
S_{31}	S_{10}	S_{50}	1
S_{41}	S_{21}	S_{41}	1
S_{50}	S_{30}	S_{21}	0
S_{51}	S_{30}	S_{21}	1

Tabelo prehajanja stanj Mooreovega avtomata lahko preoblikujemo, tako da uvedemo nove oznake stanj ($S_1 = S_{10}$, $S_2 = S_{21}$, itd.). Z novimi oznakami stanj potem zapišemo končno tabelo prehajanja stanj.

PRIMER 8.2 Pretvorba Mealyjevega avtomata v Mooreov avtomat.

ME	x_1	x_2
S_1	S_2, 1	S_3, 0
S_2	S_5, 1	S_3, 1
S_3	S_1, 0	S_5, 0
S_4	S_2, 1	S_3, -
S_5	S_3, 0	S_2, 1

Mealyjev avtomat nikoli ne preide v stanje S_4, kar lahko pomeni da je to začetno stanje avtomata. Torej stanje S_4 zapišemo v Mooreovem avtomatu brez izhodne črke.

MO	x_1	x_2	y
S_{10}	S_{21}	S_{30}	0
S_{21}	S_{51}	S_{31}	1
S_{30}	S_{10}	S_{50}	0
S_{31}	S_{10}	S_{50}	1
S_{3-}	S_{10}	S_{50}	-
S_4	S_{21}	S_{3-}	-
S_{50}	S_{30}	S_{21}	0
S_{51}	S_{30}	S_{21}	1

8.1.2 Pretvorba Moore (MO)→ Mealy (ME)

Pri pretvorbi ostane razpredelnica stanj nespremenjena (število stanj je v obeh avtomatih enako). Izhodne črke za Mealyjev avtomat določimo tako, da stanju dodamo njemu ustrezajočo izhodno črko.

PRIMER 8.3 Pretvorba Mooreovega avtomata v Mealyjev avtomat.

MO	x_1	x_2	y
S_1	S_2	S_3	0
S_2	S_5	S_3	1
S_3	S_1	S_5	0
S_4	S_2	S_4	1
S_5	S_3	S_3	0

Število stanj Mealyjevega avtomata bo enako številu stanj Mooreovega avtomata. V tabelo Mealyjevega avtomata prepišemo stanja Mooreovega in jim dodamo uztezno izhodno črko iz osnovne tabele.

ME	x_1	x_2
S_1	S_2, 1	S_3, 0
S_2	S_5, 0	S_3, 0
S_3	S_1, 0	S_5, 0
S_4	S_2, 1	S_4, 1
S_5	S_3, 0	S_3, 0

PRIMER 8.4 Pretvorba Mooreovega avtomata v Mealyjev avtomat.

MO	x_1	x_2	y
S_1	S_2	S_1	1
S_2	S_3	S_3	1
S_3	S_1	-	0
S_4	S_5	S_4	-
S_5	S_3	S_5	0

ME	x_1	x_2
S_1	S_2, 1	S_1, 1
S_2	S_3, 0	S_3, 0
S_3	S_1, 1	-
S_4	S_5, 0	S_4,-
S_5	S_3, 0	S_5, 0

8.2 Minimizacija avtomatov

V postopku minimizacije avtomatov je potrebno ločiti dve obliki avtomatov:

- **deterministične (popolne) avtomate** - avtomat je opredeljen za vse pare S_i, x_j
- **nedeterministične (delne) avtomate** - obstojajo pari S_i, x_j, kjer funkcija prehodov ni določena (pri minimizaciji je neopredeljen prehod uporabljen kot redundanca)

8.2.1 Minimizacija determinističnih avtomatov

Postopek minimizacije:

1. Stanja avtomata so grupirana glede na izhodne črke v grupe označene z $G_{i,j}$, kjer je i - indeks grupe in j - korak postopka.

V grupirano tabelo je poleg ustreznih stanj vpisan še indeks grupe - S(t+1)/i.

2. Grupe $G_{i,j}$ se ponovno deli v $G_{k,j+1}$ in ta delitev se ponavlja dokler niso v vsaki grupi stanja, ki imajo enake indekse grup pri vseh vhodnih črkah.

Vsaki grupi se priredi stanje poenostavljenega avtomata in izpiše se minimizirana razpredelnica za ustrezen avtomat.

PRIMER 8.5 Minimizirajte podan Mealyjev avtomat.

ME	x_1	x_2
S_1	S_8, 0	S_7, 1
S_2	S_3, 0	S_5, 0
S_3	S_2, 0	S_1, 0
S_4	S_5, 1	S_8, 0
S_5	S_8, 0	S_4, 1
S_6	S_5, 1	S_3, 0
S_7	S_1, 1	S_8, 0
S_8	S_4, 0	S_6, 1

Zapišemo stanja v grupe na osnovi združevanja glede na izhodne črke.

		x_1	x_2
	S_1	$S_8/1$	$S_7/3$
	S_5	$S_8/1$	$S_4/3$
$G_{1,0}$	S_8	$S_4/3$	$S_6/3$
	S_2	$S_3/2$	$S_5/1$
$G_{2,0}$	S_3	$S_2/2$	$S_1/1$
	S_4	$S_5/1$	$S_8/1$
	S_6	$S_5/1$	$S_3/2$
$G_{3,0}$	S_7	$S_1/1$	$S_8/1$

		x_1	x_2
	S_1	$S_8/2$	$S_7/4$
$G_{1,1}$	S_5	$S_8/2$	$S_4/4$
$G_{2,1}$	S_8	$S_4/4$	$S_6/5$
	S_2	$S_3/3$	$S_5/1$
$G_{3,1}$	S_3	$S_2/3$	$S_1/1$
	S_4	$S_5/1$	$S_8/2$
$G_{4,1}$	S_7	$S_1/1$	$S_8/2$
$G_{5,1}$	S_6	$S_5/1$	$S_3/3$

V vseh grupah so prehodi enolično določeni, zato je postopek minimizacije končan in grupe preimenujemo v stanja minimalnega avtomata.

$S_1 = G_{1,1}$ $S_2 = G_{2,1}$ $S_3 = G_{3,1}$ $S_4 = G_{4,1}$ $S_5 = G_{5,1}$

V tabelo vpišemo prehode stanj minimalnega avtomata in jim dodamo izhodne črke iz podane tabele Mealyjevega avtomata.

ME	x_1	x_2
S_1	S_2,0	S_4,1
S_2	S_4,0	S_5,1
S_3	S_3,0	S_1,0
S_4	S_1,1	S_2,0
S_5	S_1,1	S_3,0

PRIMER 8.6 Minimizirajte podan Mooreov avtomat.

MO	x_1	x_2	y
S_1	S_5	S_3	0
S_2	S_5	S_3	1
S_3	S_1	S_5	0
S_4	S_2	S_4	1
S_5	S_3	S_3	0

		x_1	x_2
	S_1	$S_5/1$	$S_3/1$
	S_3	$S_1/1$	$S_5/1$
$G_{1,0}$	S_5	$S_3/1$	$S_3/1$
	S_2	$S_5/1$	$S_3/1$
$G_{2,0}$	S_4	$S_2/2$	$S_4/2$

		x_1	x_2
	S_1	$S_5/1$	$S_3/1$
	S_3	$S_1/1$	$S_5/1$
$G_{1,1}$	S_5	$S_3/1$	$S_3/1$
$G_{2,1}$	S_2	$S_5/1$	$S_3/1$
$G_{3,1}$	S_4	$S_2/2$	$S_4/3$

Zapišemo stanja minimalnega avtomata:

$S_1 = G_{1,1}$ $\quad$ $S_2 = G_{2,1}$ $\quad$ $S_3 = G_{3,1}$

MO	x_1	x_2	y
S_1	S_1	S_1	0
S_2	S_1	S_1	1
S_3	S_2	S_3	1

8.2.2 Minimizacija nedeterminističnih avtomatov

Postopek:

Tvorimo pare psevdokompatibilnih stanj: dve stanji tvorita psevdokompatibilen par, če se ujemata v izhodnih črkah za vhodno abecedo X oz. se razlikujeta v izhodnih črkah le takrat, ko je katera od izhodnih črk prazna črka (-).

Za psevdokompatibilne pare določimo tabelo prehajanja stanj pri vhodni abecedi X. Če kakšen par vodi v disjunkcijo dveh stanj, ki ni par psevdokompatibilnih stanj, ga izločimo in izločimo tudi par, ki vsebuje tak izločen par. Izločanje parov je končano, ko vsi pari prehajajo v katerega od njih, ali pa v nedoločeno stanje (-). Ti pari so pravi kompatibilni pari.

Prave kompatibilne pare združimo v maksimalne kompatibilne razrede, kjer moramo

upoštevati naslednje pogoje:
a. minimalnost, ki definira takšno določanje maksimalnih razredov, da ni možno združevanje na kakšen drug način, ki pripelje do minimalnejšega avtomata
b. polnost, ki zahteva vsebovanost vseh stanj v maksimalnih kompatibilnih razredih
c. zaprtost, ki pravi da morajo stanja enega razreda v času (t) prehajati v stanja le enega razreda v času (t+1)

Stanje ki ne sodi v nobenega od kompatibilnih parov tvori samo zase kompatibilni razred.

PRIMER 8.7 Minimizirajte podan nedeterminističen Mooreov avtomat.

MO	x_1	x_2	y
S_1	-	S_1	1
S_2	S_4	S_2	1
S_3	S_5	S_3	-
S_4	S_1	-	0
S_5	S_1	S_2	-

Zapišemo tabelo psevdokompatibilnih parov:

	x_1	x_2
1,2	4	1,2
1,3	5	1,3
1,5	1	1,2
2,3	4,5	2,3
~~2,5~~	~~1,4~~	~~2~~
3,4	1,5	3
3,5	1,5	2,3
4,5	1	2

- izločimo vrstico kompatibilnega para 2,5 ki prehaja v par 1,4 ki ni psevdokompatibilen par
- združimo kompatibilna para 1,2 in 1,3 v maksimalen razred in ugotovimo da ta razred zadošča pogoju zaprtosti. - maksimalni razred (1,2,3) ob vhodni črki x_1 prehaja v stanji 4,5, kar pogojuje definicijo para 4,5 kot drugi maksimalni razred, ki tudi zadošča pogoju zaprtosti. Istočasno je izpolnjen tudi pogoj polnosti, ker so v maksimalnih razredih vsa stanja avtomata.
- pogoj minimalnosti je tudi izpolnjen, zato ker glede na definicijo dveh izhodnih črk 0 in 1 ni možno dobiti v minimalni obliki avtomata z manj kot dvema stanjema
Iz maksimalnih kompatibilnih razredov tvorimo stanji minimalnega avtomata, kjer je S_1 = (1,2,3), S_2 = (4,5) in zapišemo minimalno obliko Mooreovega avtomata.

MO	x_1	x_2	y
S_1	S_2	S_1	1
S_2	S_1	S_1	0

PRIMER 8.8 Minimizirajte podan nedeterminističen Mealyjev avtomat.

ME	x_1	x_2
S_1	S_4, -	S_3, 1
S_2	S_2, 1	S_1, 0
S_3	S_6, 1	S_3, 0
S_4	-	S_6,1
S_5	S_5, 1	S_3, 1
S_6	S_5, 1	-

Zapišemo tabelo psevdokompatibilnih parov:

	x_1	x_2
1,4	4	3,6
1,5	4,5	3
1,6	4,5	3
~~2,3~~	~~2,6~~	~~1,3~~
~~2,6~~	~~2,5~~	~~1~~
3,6	5,6	3
4,5	5	3,6
5,6	5	3

- para 2,3 in 2,6 izločimo, ker prehajata v para 1,3 in 2,5, ki nista kompatibilna para
- stanja 1,4,5,6 združimo v en maksimalen razred, ker je izpolnjen pogoj zaprtosti in ob vhodu x_2 pogojuje določitev para 3,6 kot drugi maksimalen razred.
- stanje 2 v naboru kompatibilnih parov ni definirano, zato za izpolnjenost pogoja polnosti tvori svoj razred

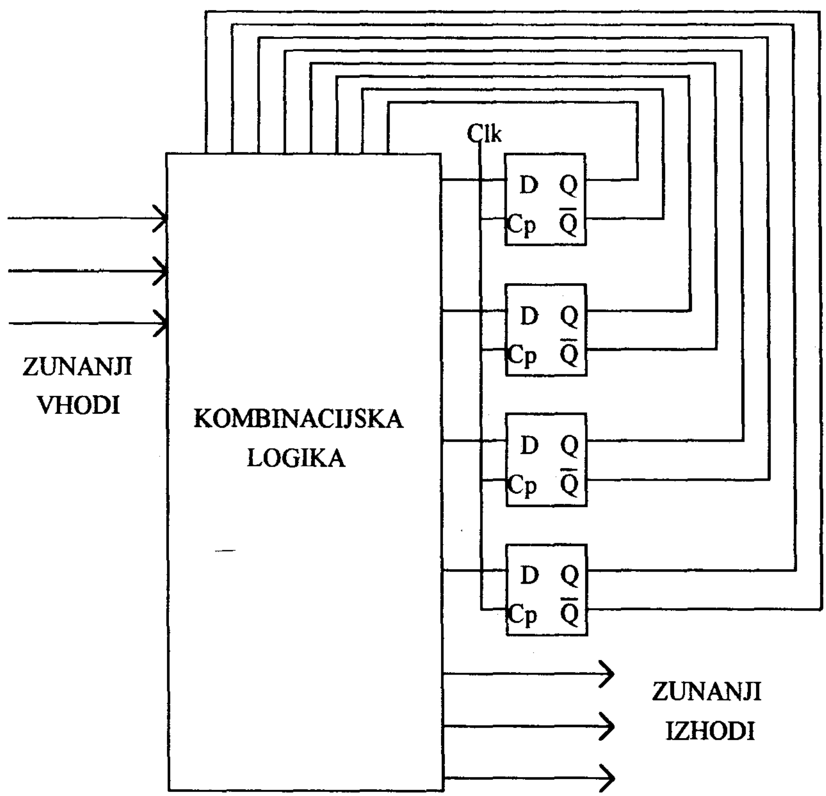
Iz maksimalnih kompatibilnih razredov tvorimo stanja minimalnega avtomata, kjer so S_1 = (1,4,5,6), S_2 = (3,6), S_3 = (2) in zapišemo minimalno obliko Mealyjevega avtomata.

ME	x_1	x_2
S_1	S_1, 1	S_2, 1
S_2	S_1, 1	S_2, 0
S_3	S_3, 1	S_1, 0

8.3 Realizacija končnega avtomata

Avtomati (sekvenčni stroji) predstavljajo pomemben razred sekvenčnih digitalnih logičnih vezij, ki jih lahko označimo z naslednjimi lastnostmi:
1. Ima dobro določeno kombinacijsko logiko.
2. Ima sinhronski register, katerega logični izhodi so uporabljeni direktno, ali pa so ponovno uporabljeni kot vhodi v kombinacijsko logiko.
3. Kombinacijska logika lahko sprejme kot vhod zunanje vhode, ali povratne signale iz registra, ali pa oboje. Generira izhode, ki so dosegljivi preko zunanjih izhodov in/ali so uporabljeni kot vhodi v register.

Osnovne komponente avtomata vidimo na spodnji sliki.

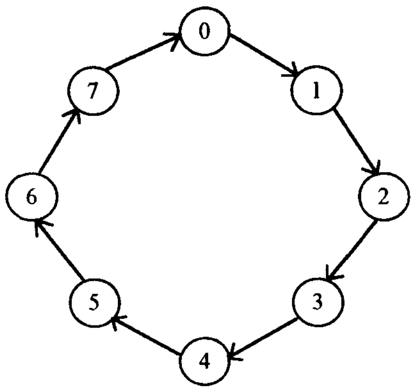


Namen kombinacijske logike je, da zagotavlja zahtevane ukaze za delovanje avtomata, predno nastopi urin impulz. Ločitev kombinacijske logike od registerskega dela zagotavlja prevedbo načrtovanja avtomata na čisti kombinacijski logični problem. Pomembna značilnost avtomata je sinhronizacija logike v registru, ki zagotavlja, da časovne razlike elementov v kombinacijskem vezju ne vplivajo na delovanje.

8.3.1 Postopek za realizacijo končnega avtomata z D pomnilnimi celicami

Pri realizaciji avtomata imamo trenutno stanje sistema shranjeno v registru. Kombinacijska logika sprejme podatek o trenutnem stanju avtomata kot vhod in generira naslednje stanje, ki je prisotno na registerskih D vhodih. Ob pojavu urinega impulza se ta podatek shrani v register in postane novo trenutno stanje. Proces spreminjanja stanj se ponavlja. Ker so podatki o naslednjem stanju med urinimi impulzi konstantni, lahko uporabimo pravilnostno tabelo za določitev naslednjega stanja kot funkcijo trenutnega stanja in zunanjih vhodov.

PRIMER 8.8 Realizacija sinhronskega števca po mod 8, ki ga predstavimo z diagramom prehajanja stanj. Avtomat, ki ga realiziramo nima zunanjih vhodov, izhodi avtomata pa so kar stanja avtomata (izhodi registerskega dela).



1.Število potrebnih pomnilnih celic N pri avtomatu z M stanji je

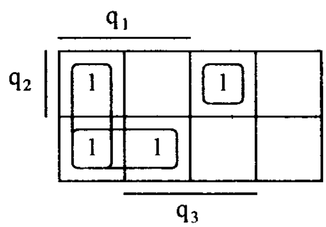
$$N \geq \log_2 M \rightarrow 2^N \geq M$$

Za M = 8 dobimo iz $N \geq \log_2 M \rightarrow 2^3 = 8$, da so potrebne 3 pomnilne celice v realizaciji.

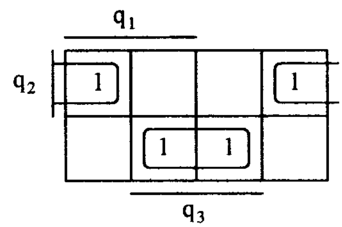
2. Zapis binarne aplikacijske tabele, ki sledi iz diagrama prehajanja stanj števca.

$q_1(t)$	$q_2(t)$	$q_3(t)$	$q_1(t+1)$	$q_2(t+1)$	$q_3(t+1)$
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

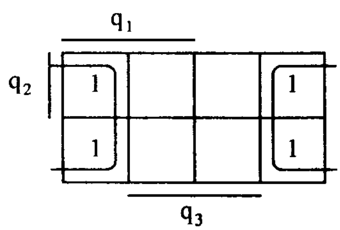
3. Določitev krmilnih funkcij za D pomnilne celice pomeni, da moramo zapisati funkcije naslednjega stanja iz tabele v odvisnosti od trenutnega stanja avtomata. Zapišemo funkcije $D_1 = q_1(t+1)$, $D_2 = q_2(t+1)$, $D_3 = q_3(t+1)$ v Veitchev diagram in poiščemo minimalno obliko.



$D_1 = q_1\overline{q}_3 \vee q_1\overline{q}_2 \vee \overline{q}_1 q_2 q_3$



$D_2 = q_2\overline{q}_3 \vee \overline{q}_2 q_3 = q_2 \nabla q_3$



$D_3 = \overline{q}_3$

Dobili smo vhodne funkcije v D pomnilne celice in tako je kombinacijsko vezje avtomata popolnoma določeno. Sama realizacija je potem odvisna od tega katere logične gradnike bomo uporabili.

PRIMER 8.9 Realizacija sinhronskega UP/DOWN števca po mod 4, ki ga predstavimo s tabelo prehajanja stanj. Avtomat, ki ga realiziramo ima en zunanji vhod, ki določa način štetja. Izhodi avtomata pa so kar izhodi pomnilnih celic (registerskega dela avtomata).

	UP	DOWN
0	1	3
1	2	0
2	3	1
3	0	2

1.Število potrebnih pomnilnih celic N pri avtomatu z M stanji je

$N \geq \log_2 M \rightarrow 2^N \geq M$

Za M = 4 dobimo iz $N \geq \log_2 M \rightarrow 2^2 = 4$, da sta potrebni 2 pomnilni celici v realizaciji.

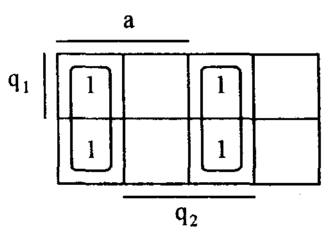
2. Zunanji vhod določa krmilno operacijo števca UP/DOWN, ki ga kodiramo:

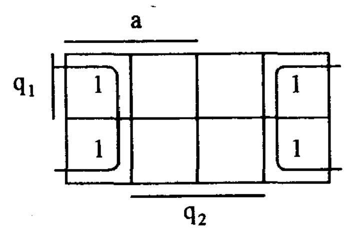
a	
0	UP
1	DOWN

3. Zapis binarne aplikacijske tabele, ki sledi iz tabele prehajanja stanj števca in kodirne tabele.

a	$q_1(t)$	$q_2(t)$	$q_1(t+1)$	$q_2(t+1)$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

4. Določitev krmilnih funkcij za D pomnilne celice pomeni, da moramo zapisati funkcije naslednjega stanja iz tabele v odvisnosti od trenutnega stanja avtomata. Zapišemo funkcije $D_1 = q_1(t+1)$, $D_2 = q_2(t+1)$ v Veitchev diagram in poiščemo minimalno obliko.





$D_1 = a\bar{q}_2 \vee \bar{a}\, q_2 = a \nabla q_2$

$D_2 = \bar{q}_2$

Dobili smo vhodne funkcije v D pomnilne celice in tako je kombinacijsko vezje avtomata popolnoma določeno. Sama realizacija je zelo enostavna, če uporabimo EX-OR operator.

8.3.2 Postopek za realizacijo končnega avtomata z JK pomnilnimi celicami

Pri realizaciji avtomatov lahko pogosto izbiramo med uporabo D in JK pomnilnih celic. Izbira zavisi od trenutne aplikacije, kar pomeni, da je ena od rešitev ugodnejša od druge. V naslednjem primeru bomo videli bistvene razlike v načrtovanju z izbranimi pomnilnimi celicami.

PRIMER 8.10 Realizacija števca v Grayevi kodi s podano sekvenco za JK pomnilne celice.

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

↓ sekvenca se ponovi

Zapišimo binarno aplikacijsko tabelo za prehajanje stanj avtomata, ki sledi iz definicije števca. Vhodne kombinacije oz. stanja, ki v števcu niso upoštevana so v tabeli neopredeljene (redundančne). Avtomat ima izhod definiran s stanji in nima nobenega zunanjega vhoda.

trenutno stanje				naslednje stanje			
D	**C**	**B**	**A**	**D**	**C**	**B**	**A**
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	1
0	0	1	0	×	×	×	×
0	0	1	1	0	1	1	0
0	1	0	0	×	×	×	×
0	1	0	1	×	×	×	×
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	1	0	0
1	0	1	0	×	×	×	×
1	0	1	1	×	×	×	×
1	1	0	0	1	1	1	0
1	1	0	1	×	×	×	×
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

1. Število potrebnih pomnilnih celic N pri avtomatu z M stanji je

$$N \geq \log_2 M \rightarrow 2^N \geq M$$

Za M = 10 dobimo iz $N \geq \log_2 M \rightarrow 2^4 \geq 10$, da so potrebne 4 pomnilne celice v realizaciji.

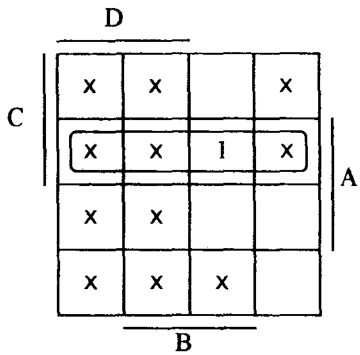
2. Za realizacijo vezja potrebujemo tabelo prehodov za JK pomnilno celico, na osnovi katere bomo v binarno aplikacijsko tabelo vpisali vhodne funkcije.

$Q(t) \rightarrow Q(t+1)$	J	K
$0 \rightarrow 0$	0	×
$0 \rightarrow 1$	1	×
$1 \rightarrow 0$	×	1
$1 \rightarrow 1$	×	0

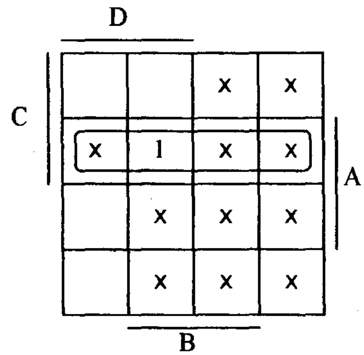
3. Vpis vhodnih funkcij v binarno aplikacijsko tabelo.

trenutno stanje				naslednje stanje											
D	C	B	A	D	C	B	A	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	0	0	1	0	×	0	×	0	×	1	×
0	0	0	1	0	0	1	1	0	×	0	×	1	×	×	0
0	0	1	0	×	×	×	×	×	×	×	×	×	×	×	×
0	0	1	1	0	1	1	0	0	×	1	×	×	0	×	1
0	1	0	0	×	×	×	×	×	×	×	×	×	×	×	×
0	1	0	1	×	×	×	×	×	×	×	×	×	×	×	×
0	1	1	0	0	1	1	1	0	×	×	0	×	0	1	×
0	1	1	1	1	0	0	0	1	×	×	1	×	1	×	1
1	0	0	0	1	0	0	1	×	0	0	×	0	×	1	×
1	0	0	1	1	1	0	0	×	0	1	×	0	×	×	1
1	0	1	0	×	×	×	×	×	×	×	×	×	×	×	×
1	0	1	1	×	×	×	×	×	×	×	×	×	×	×	×
1	1	0	0	1	1	1	0	×	0	×	0	1	×	0	×
1	1	0	1	×	×	×	×	×	×	×	×	×	×	×	×
1	1	1	0	1	1	1	1	×	0	×	0	×	0	1	×
1	1	1	1	0	0	0	0	×	1	×	1	×	1	×	1

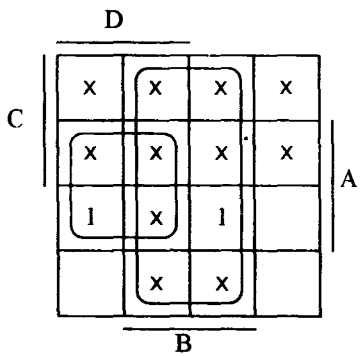
Zapišemo vhodne funkcije za JK pomnilne celice v Veitchev diagram in poiščemo minimalne oblike.



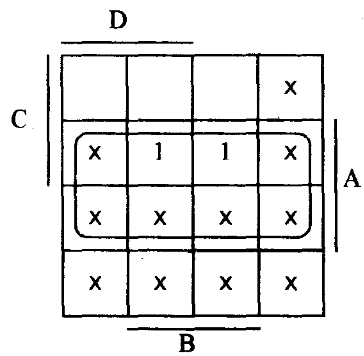
$J_D = CA$



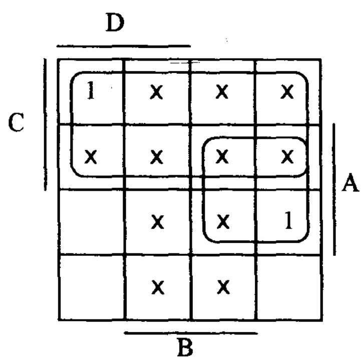
$K_D = CA$



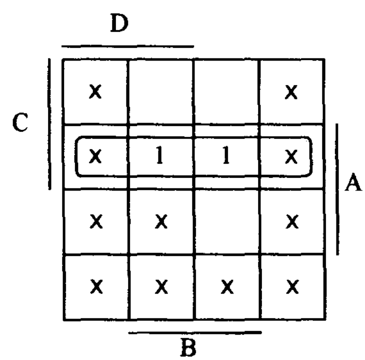
$J_C = B \vee DA$



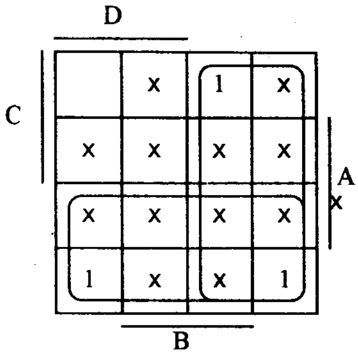
$K_C = A$

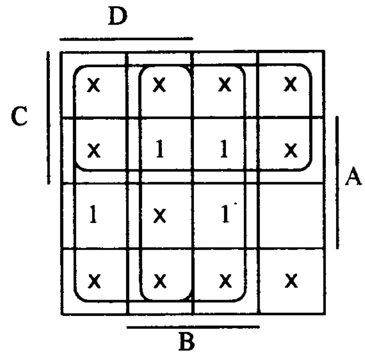


$J_B = C \vee \overline{D}A$



$K_B = CA$





$J_A = \overline{D} \vee \overline{C}$ $K_A = D \vee C \vee B$

Dobili smo zapis vhodnih funkcij v minimalni obliki. Realizacija vezja je odvisna od tega katere gradnike bomo uporabili za kombinacijsko vezje.

Literatura:

A.Dobnikar, Zapiski predavanj za Preklopna vezja in strukture

A:D:Friedman, Fundamentals of Logic Design and Switching Theory, Computer Science Press, Inc., USA 1986

S.Muroga, Logic Design and Switching Theory, John & Sons, Inc.,USA 1986

J.E.Palmer, D.E.Perlman, Introduction to Digital Systems, Schaum's Outline Series McGraw-Hill. Inc., 1993

J.Virant, Teorija preklopnih vezij, Zal. FE v Ljubljani, 1979

J.Virant, Logične osnove odločanja in pomnjenja v računalniških sistemih, Zal. FER v Ljubljani, 1990

Priročnik za vaje Preklopna vezja vsebuje ... pomnenja v preklopnih strukturah in siste... binarnega številskega sistema in Booleova a... preklopnih vezij. V nadaljevanju so opisane preklopne funkcije in postopki minimizacije, ki jih srečamo v odločitvenih vezjih. Realizacija preklopnih funkcij je prikazana z različnimi tipi gradnikov v SSI, MSI, LSI in VLSI tehnologijah. V drugem delu so podrobneje opisana sekvenčna preklopna vezja, kjer je funkciji odločanja dodana funkcija pomnenja. V zadnjem delu priročnika je vključenih še nekaj osnovnih znanj in primerov iz teorije avtomatov.

Avtor:

Mag. Mira Trebar, dipl. ing. je asistent na FER. Do sedaj je imela dve izdaji Priročnika za vaje iz Preklopnih vezij in struktur za potrebe študentov. V okviru Laboratorija za računalniške strukture in sisteme ter Laboratorija za adaptivne sisteme in paralelno procesiranje je sodelovala pri raziskovalnih nalogah za industrijo in raziskovalno skupnost. Iz opravljenega raziskovalnega dela je soavtor 9 člankov in referatov doma in v tujini.