4. Minimizacija Booleovih izraza

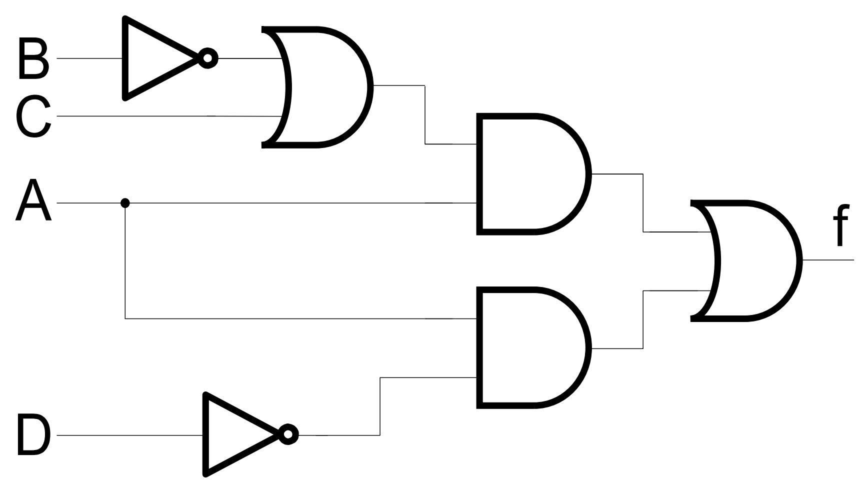
4. Minimizacija Booleovih izraza

- minimum Booleove funkcije
- K tablice
- minimizacija K tablicama
- vremenski hazard
- Quine-McCluskeyeva metoda
- minimizacija višeizlazne funkcije
- Quine-McCluskey za višeizlazne funkcije

Minimum Booleove funkcije

- podsjetiti se:
  - Booleova funkcija je opis digitalnog sklopa:
    - operator ⇔ osnovni logički sklop
    - izraz koji utvrđuje Booleovu funkciju ⇔ sklop

Primjer : $f = A \cdot (\overline{B} + C) + A\overline{D}$



Minimum Booleove funkcije

- želja:
  - postići minimalno ostvarenje dane Booleove funkcije:
    - najjednostavniji sklop
      ~ niz pogodnosti
    - "jednostavan" sklop = ?
      ~ kriteriji jednostavnosti
    - mjere složenosti sklopa?
  - pojednostavljivanje izraza
    ~ pojednostavljivanje sklopa:
    - tehnički razlozi
      ~ potrošnja, disipacija, …
    - ekonomski razlozi
      ~ cijena sklopova, prostor na pločici, …

Minimum Booleove funkcije

- mjere složenosti digitalnog sklopa:
  - brzina rada ~ broj razina "logike"
  - broj utrošenih primitivnih sklopova
    - bez ograničenja
    - izvedba *u dvije razine*
  - broj utrošenih primitivnih sklopova
+ ukupan broj ulaza u logičke sklopove
    - bez ograničenja
    - izvedba *u dvije razine*

Minimum Booleove funkcije

- minimizacija (engl. minimization)
  ~ pronaći izraz koji minimizira
  odabranu mjeru složenosti:

"za zadanu funkciju od n varijabli iz skupa 2^{2^n} njih, odrediti onaj izraz, unutar velikog broja ekvivalentnih, koji će zadovoljiti neke od kriterija jednostavnosti"

Minimum Booleove funkcije

- kriteriji jednostavnosti kontradiktorni
  ~ uobičajeno u inženjerskoj praksi!
  - najveća brzina rada sklopa
    ~ funkcija drugog reda :
    dvije razine logike (ILI-I, NI-NI, ...)
  - najjeftinije ostvarenje
    ~ min broj standardnih sklopova
    ili izvoda/kućišta standardnih modula
    - eventualni porast broja razina logike
      ~ zapis "funkcija višeg reda"
    - faktorizacija
      ~ dekompozicija u češće korištene
      komponentne funkcije
    - vrijeme propagacije signala nije minimalno !

Minimum Booleove funkcije

- standardni postupak minimizacije
  ~ primjena na funkcije drugog reda

  "Neki se izraz drugog reda u obliku sume produkata smatra minimalnim – minimiziranim – ako ne postoji:
  - niti jedan drugi ekvivalentni izraz s manje produkata,
  - niti jedan drugi ekvivalentni izraz s istim brojem produkata, ali manjim brojem literala."

  literal = {varijabla | komplement}

Minimum Booleove funkcije

- neke definicije (1):
  - implikant, i_i:
    - produkt u zapisu funkcije kao sume produkata
    - "implicira" f = 1

$$f = B\overline{C}D + BCD + A\overline{C}D$$

$$i_1 = B\overline{C}D, i_2 = BCD, i_3 = A\overline{C}D$$

  - primarni implikant (primarni član), pi_i:
    ~ implikant koji se ne može kombinirati u drugi implikant s manjim brojem literala

$$f = B\overline{C}D + BCD + A\overline{C}D = BD + A\overline{C}D$$

$$pi_1 = BD, pi_2 = A\overline{C}D$$

Minimum Booleove funkcije

- neke definicije (2):
  - *bitni primarni implikant*
    ~ primarni implikant koji jedini
    prekriva (engl. cover) neki m_i
  - *potpuna suma* (engl. complete sum)
    ~ suma *svih* primarnih implikanata funkcije, Σpi_i
  - *minimalna suma* ≡ minimalno prekrivanje
    ~ suma primarnih implikanata koja prekriva (sadrži)
    sve minterme funkcije uz *minimalni* broj članova

Minimum Booleove funkcije

- sintaksne manipulacije Booleovog izraza
  ~ algebarska minimizacija :
  - transformacija funkcije zamjenom jednog njenog oblika (izraza) drugim, uzastopnom primjenom postulata i teorema Booleove algebre
  - ne postoji sustavan postupak koji vodi do minimuma

Minimum Booleove funkcije

Primjer : $f(A,B,C) = B\overline{C}(\overline{C} + \overline{C}A) + (\overline{A} + \overline{C})(\overline{A}B + \overline{A}C)$

$$
\begin{aligned}
f(A,B,C) &= B\overline{C}(\overline{C} + \overline{C}A) + (\overline{A} + \overline{C})(\overline{A}B + \overline{A}C) \\
&= B\overline{C} \cdot \overline{C} \cdot (1 + A) + \overline{A} \cdot (\overline{A} + \overline{C})(B + C) \\
&= B\overline{C} + \overline{A} \cdot (B + C) \\
&= B\overline{C} + \overline{A}B + \overline{A}C \\
&= B\overline{C} + \overline{A}B \cdot (C + \overline{C}) + \overline{A}C \\
&= (B\overline{C} + \overline{A}B\overline{C}) + (\overline{A}C + \overline{A}BC) \\
&= \overline{A}C(1 + B) + B\overline{C}(1 + \overline{A}) \\
&= \overline{A}C \cdot 1 + B\overline{C} \cdot 1 \\
&= \overline{A}C + B\overline{C}
\end{aligned}
$$

K tablice

- grafički prikaz Booleovih funkcija:
  - tablica u 2-dimenzijskom obliku
  - polja
    ~ standardni članovi (produkti/sume)
  - "razlika" grafički susjednih polja u samo jednoj varijabli!

A	B	f
0	0	α_0
0	1	α_1
1	0	α_2
1	1	α_3

$\Rightarrow$

f(A,B)

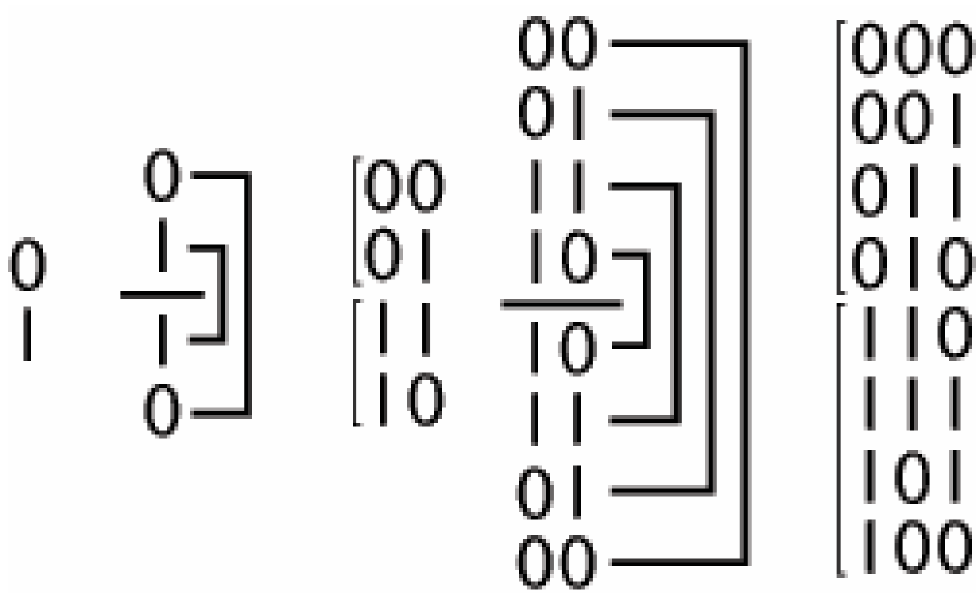
B \ A	0	1
0	α_0	α_2
1	α_1	α_3

K tablice

- *K-tablice* (Karnaughove tablice), M. Karnaugh, 1953:
  - grafičke strukture s 2^n polja za prikaz
    $f(x_1, x_2, \ldots, x_n)$
  - označavanje polja
    ~ "pravokutne koordinate", Grayev kod ($d_{min} = 1$)
  - minimizacija
    ~ "grupiranje" polja:
    temeljeno na ljudskoj sposobnosti
    raspoznavanja uzoraka (1 i 0)
  - K-tablice za n > 2 varijable
    ~ simetrija oko jedne stranice, superpozicija
  - praktična primjena: $n \leq 6$

Grayev kod

- podsjetnik:



K tablice

- izgradnja K tablice:

f(A,B)

B \ A	0	1
0	0	2
1	1	3

f(A,B,C)

C \ AB	00	01	11	10
0	0	2	6	4
1	1	3	7	5

f(A,B,C,D)

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

f(A,B,C,D,E)

DE \ ABC	000	001	011	010	110	111	101	100
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

f(A,B,C,D,E)

DE \ ABC	000	010	110	100	001	011	111	101
00	0	8	24	16	4	12	28	20
01	1	9	25	17	5	13	29	21
11	3	11	27	19	7	15	31	23
10	2	10	26	18	6	14	30	22

K tablice

- susjednost polja:

f(A,B,C,D)

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

$$13 = 1101 \equiv AB\overline{C}D \quad \rightarrow \quad \begin{array}{lcl} 12 = 1100 \equiv AB\overline{C}\overline{D} & : & D \equiv 2^0 \\ 15 = 1111 \equiv ABCD & : & C \equiv 2^1 \\ 09 = 1001 \equiv A\overline{B}\overline{C}D & : & B \equiv 2^2 \\ 05 = 0101 \equiv \overline{A}B\overline{C}D & : & A \equiv 2^3 \end{array}$$

K tablice

- upisivanje funkcija u K-tablice:
  - funkcija u obliku sume minterma, Σm_i:
    1 za svaki m_i
  - funkcija u obliku produkta maksterma, ΠM_i:
    0 za svaki M_i, ostalo su 1 (0 se ne pišu!)
  - *nepotpuno specificirane funkcije*
    (engl. incompletely specified functions):
    - parcijalne funkcije
    - neke kombinacije argumenata se *ne pojavljuju*:
      - funkcijska vrijednost *nije specificirana*, X (engl. don't care)
      - X se interpretiraju onako *kako najbolje odgovara pri minimizaciji! ("džoker")*

K tablice

Primjer:

$$z = f(A,B,C,D)$$
$$= \sum m(4,5,13,14,15) + \sum d(1,3,7,8,12)$$

f(A,B,C,D)

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

K tablice

- prikaz "složene" Booleove funkcije
  ~ osnovne operacije nad Booleovim funkcijama:
  - jednostavno dobivanje rješenja kombiniranjem pripadnih K tablica
  - kombiniranje K tablica
    ~ kombiniranje pojedinih polja K-tablica funkcija

K tablice

Primjer : $h = f \oplus g$

f(A,B,C,D)

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

g(A,B,C,D)

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

h(A,B,C,D)

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

Minimizacija K tablicama

- postupak minimizacije za funkcije u obliku *sume produkata*:
  - "zaokruživanje" uzoraka 2^i susjednih polja s 1 ~ "eliminiranje" *i* varijabli
    - par polja: 1 varijabla (T9: simplifikacija)

$$
\begin{aligned}
f(a,b,c,\ldots) &= a\cdot\varphi(b,c,\ldots) + \overline{a}\cdot\varphi(b,c,\ldots)\\
&= (a+\overline{a})\cdot\varphi\\
&= \varphi
\end{aligned}
$$

    - četvorka polja: 2 varijable
    - osmorka polja: 3 varijable
    - itd. (ako ide ☺)

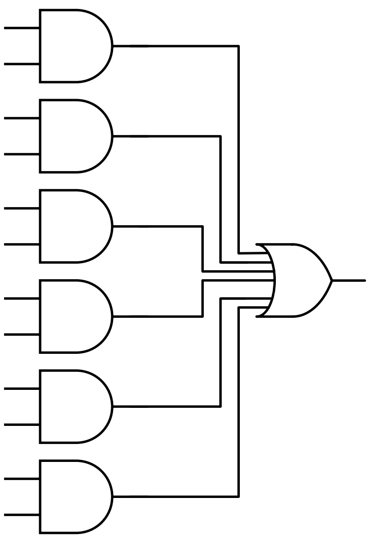
Minimizacija K tablicama

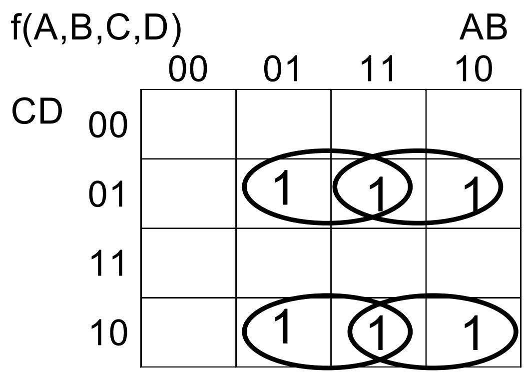
- postupak minimizacije za funkcije u obliku sume produkata :
  - "zaokruženje"
    ~ produkt, ali više nije standardni
  - inkluzivna disjunkcija zaokruženja
    ~ suma produkata (= funkcija drugog reda)
  - težnja:
    - što veći broj 1 u zaokruženju
      ~ I sklop s manjim brojem ulaza
    - što manji broj zaokruženja
      ~ manji broj I sklopova = manji broj ulaza u ILI sklop

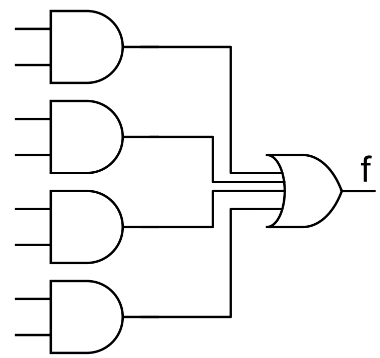
Minimizacija K tablicama

Primjer : $f(A,B,C,D)=\sum m(5,6,9,10,13,14)$

$$f(A,B,C,D)=\sum m(5,6,9,10,13,14) \quad \Rightarrow \quad f(A,B,C,D)=$$
$$=B\overline{C}D+A\overline{C}D+BC\overline{D}+AC\overline{D}$$







Minimizacija K tablicama

- postupak minimizacije *nepotpuno specificirane funkcije* u obliku sume produkata:
  - nužno je pokriti sve 1, ali ne i sve X
  - X se interpretira kao 1 (X = 1) *samo ako* se time može proširiti zaokruženje

$\Downarrow$

veće zaokruženje

~ jednostavniji Booleov izraz = jednostavniji sklop!

Minimizacija K tablicama

Primjer: $f = \sum m(2,5,15) + \sum d(0,1,3,4,7,9,13,14)$

f(A,B,C,D)

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

$$f(A,B,C,D) = \overline{A}\,\overline{B} + BD$$

Minimizacija K-tablicama

- preljevanje zaokruženja preko rubova:

f(A,B,C,D)

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

$$f(A,B,C,D) = \overline{B}\,\overline{C}D$$

Minimizacija K tablicama

- minimizacija funkcije u obliku *produkta maksterma*
  - isti postupak, samo se zaokružuju 0
  - rezultat je produkt suma
  - "čitanje" zaokruženja 0 kao sume produkata
    ~ komplement funkcije

Minimizacija K tablicama - maksterme

Primjer: $f = \prod M(5,7,15)$

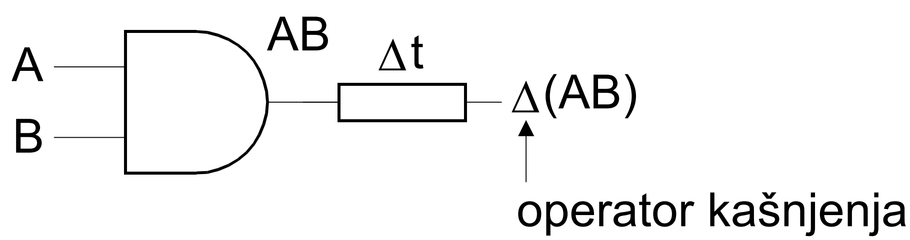
f(A,B,C,D)

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

$$f(A,B,C,D) = \left(A+\overline{B}+\overline{D}\right)\cdot\left(\overline{B}+\overline{C}+\overline{D}\right)$$

Vremenski hazard

- zapažanje:
  - *stvarni* (kombinacijski) sklopovi
    ~ svojstveno kašnjenje (t_d)!
  - promatrati ostvarenu logičku funkciju + t_d



  - *moguće* neočekivano ponašanje sklopa
    u *prijelaznoj pojavi*

Vremenski hazard

- *vremenski hazard*
  ~ neželjeni impulsi kao rezultat:
  - kašnjenja stvarnih sklopova
  - konkretnog dizajna složenijeg sklopa
    ~ struktura sklopa izražena kombinacijom jednostavnijih sklopova

Vremenski hazard

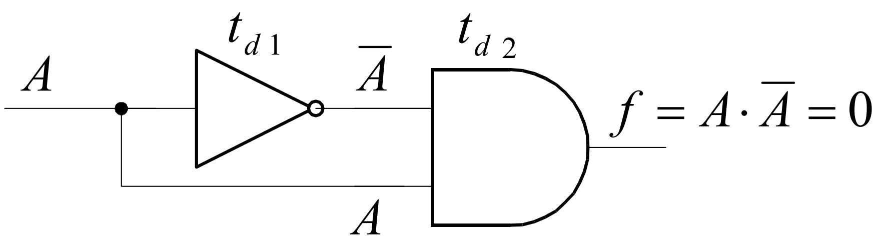
- *hazard* (rizik):

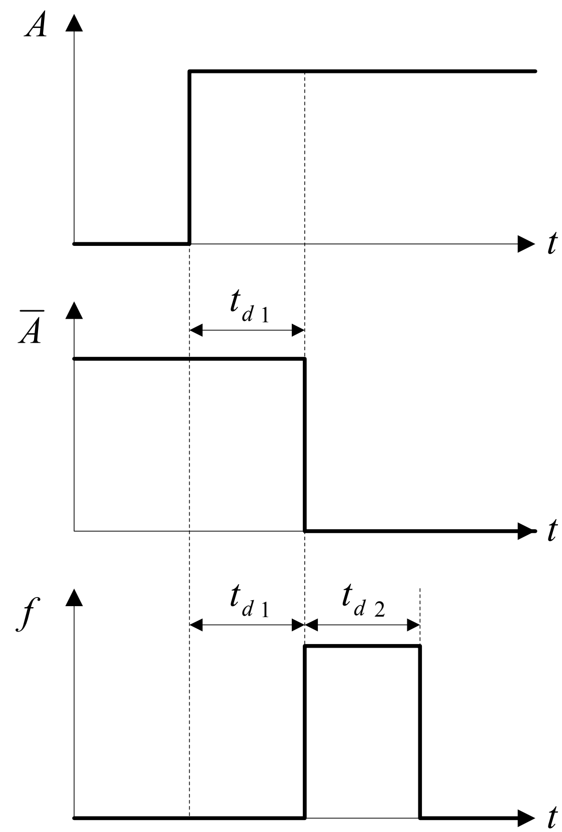
  pojava privremenog **krivog impulsa** koji u određenim slučajevima *može* prouzrokovati pogrešan rad sklopa:

  - *statički 0-hazard*:
    ~ izlaz statički u 0, a za prijelazne pojave generira se 1
  - *statički 1-hazard*:
    ~ izlaz statički u 1, a za prijelazne pojave generira se 0
  - *dinamički hazard*:
    ~ generiranje ≥ 1 impulsa pri promjeni stanja na izlazu

Vremenski hazard

Primjer: statički 0-hazard



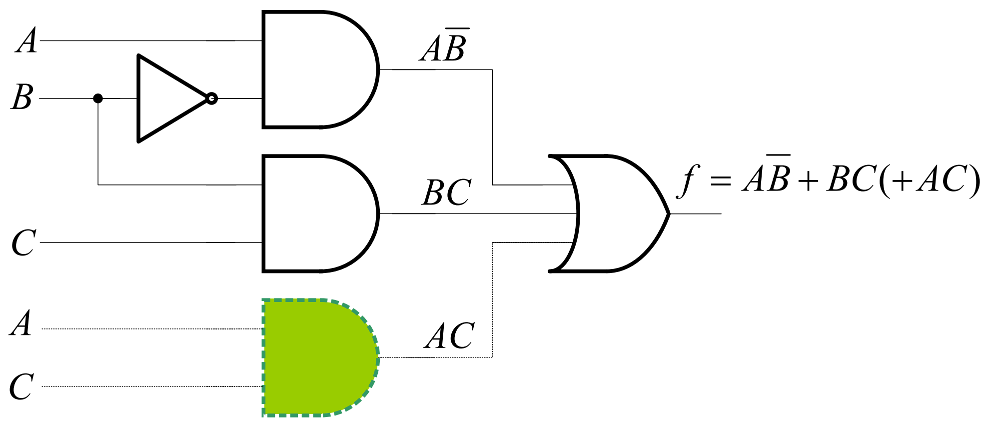


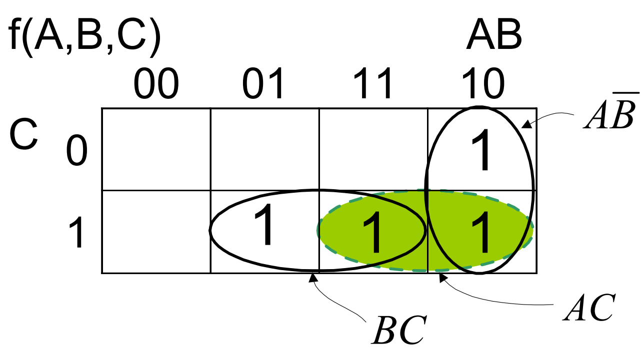
Vremenski hazard

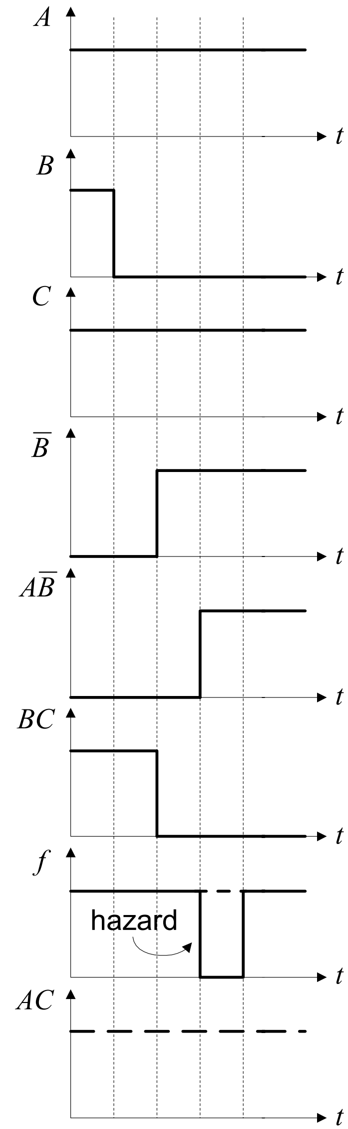
- logički hazard :
  - rezultat logičke implementacije funkcije
    ~ minimizacija Booleovog izraza!
  - statički logički hazard:
    ~ tipična pojava kad dva logička signala
    koji imaju suprotne vrijednosti (A i $\overline{A}$) poprimaju
    istu vrijednost za vrijeme prijelaznog stanja :
    - razmatrati ih kao različite signale!
    - dodati redundantni član (produkt/sumu)
  - standardno rješenje
    ~ izbjeći očitanje signala za prijelazne pojave:
    - impulsi sinkronizacije
      ~ usporavanje rada sustava!

Vremenski hazard

Primjer: $f = A\bar{B} + BC(+AC)$







Quine-McCluskeyeva metoda

- tablična metoda prikladna za minimizaciju funkcija većeg broja varijabli:
  - može se svesti na manipuliranje *indeksima* standardnih članova
  - numerički postupak
    ~ pogodan za programsku implementaciju
- W. V. Quine, 1952;
  poboljšanje: E. J. McCluskey, 1956

Quine-McCluskeyeva metoda

- potpuno specificirana funkcija u obliku sume standardnih produkata
- postupak u *dvije* faze:
  - prva faza
    ~ nalaženje *primarnih implikanata*
    (→ potpune sume):
    najveća zaokruženja u K-tablicama
  - druga faza
    ~ određivanje optimalnog (*minimalnog*) skupa primarnih implikanata (→ minimalne sume)

Quine-McCluskeyeva metoda

- prva faza:
  - svrstavanje minterma u *klase* prema broju jedinica
  - uspoređivanje elemenata *susjednih* klasa
    ~ kombiniranje elemenata koji se mogu simplificirati (T9)

    $$A \cdot \varphi + \overline{A} \cdot \varphi = \varphi \quad (*)$$

    - dobiveni produkti
      ~ klasa u novoj tablici
    - elementi koji nisu kombinirani
      ~ *primarni implikanti*
  - ponavljanje prethodnog koraka
    za elemente koji su izgubili istu varijablu
  - postupak se zaustavlja
    ~ nema više kandidata za kombiniranje

Quine-McCluskeyeva metoda

- dodaci za numerički postupak:
  - klase su susjedne
  - elementi se razlikuju za 2^k, $k = 0, 1, 2, \ldots$
  - element u višoj klasi mora biti veći
  - eliminira se varijabla 2^k

Quine-McCluskeyeva metoda

Primjer: $z = f(A,B,C,D) = \sum (1,3,5,6,9,11,12,13,14,15)$

- prva faza

	A	B	C	D
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

1	1	✓	1	1,3	(2)	✓	1	1,3,9,11	(2,8)
2	3	✓		1,5	(4)	✓		1,5,9,13	(4,8)
	5	✓		1,9	(8)	✓		1,9,3,11	(8,2)
	6	✓	2	3,11	(8)	✓		1,9,5,13	(8,4)
	9	✓		5,13	(8)	✓	2	9,11,13,15	(2,4)
	12	✓		6,14	(8)			12,14,13,15	(2,1)
3	11	✓		9,11	(2)	✓		12,13,14,15	(1,2)
	13	✓		9,13	(4)	✓		9,13,11,15	(4,2)
	14	✓		12,13	(1)	✓			
4	15	✓		12,14	(2)	✓			
			3	11,15	(4)	✓			
				13,15	(2)	✓			
				14,15	(1)	✓			

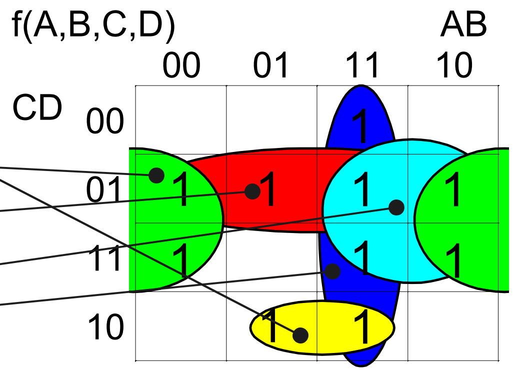
Quine-McCluskeyeva metoda

- rezultat prve faze: $z = f(A,B,C,D)$

$$= \sum (1,3,5,6,9,11,12,13,14,15)$$

- primarni članovi

6,14 (8)	$\equiv$	$BC\overline{D}$	=	a
1,3,9,11 (2,8)	$\equiv$	$\overline{B}D$	=	b
1,5,9,13 (4,8)	$\equiv$	$\overline{C}D$	=	c
9,11,13,15 (2,4)	$\equiv$	AD	=	d
12,13,14,15 (1,2)	$\equiv$	AB	=	e



Quine-McCluskeyeva metoda

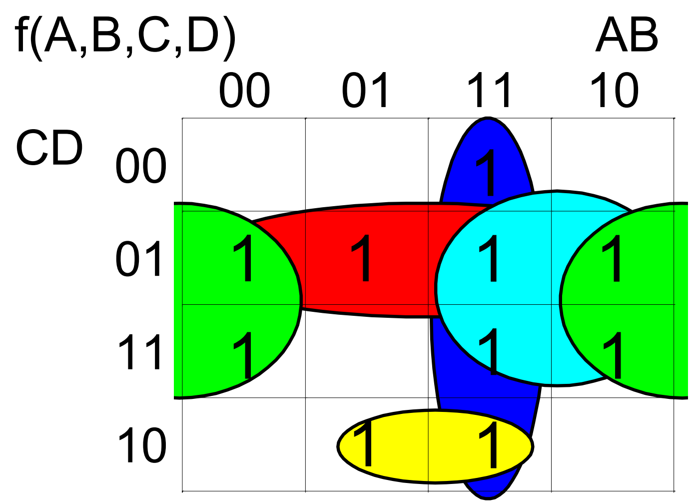
- druga faza:
  - formiranje tablice primarnih implikanata
    i označavanje prekrivanja minterma
  - nalaženje *bitnih* primarnih implikanata,
    koji *jedini prekrivaju pojedini minterm*
    ~ označiti minterme koje taj član pokriva
  - bitni primarni implikanti ulaze u minimalnu sumu
  - preostale minterme prekriti *minimalnim* podskupom
    preostalih primarnih implikanata
    ~ prednost:
    primarni implikanti s manjim brojem literala

Quine-McCluskeyeva metoda

- druga faza:

		1	3	5	6	9	11	12	13	14	15
$BC\overline{D}$	a				X					X	
$\overline{B}D$	b	X	X			X	X				
$\overline{C}D$	c	X		X		X			X		
AD	d					X	X		X		X
AB	e							X	X	X	X
		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

$$z = a + b + c + e$$
$$= BC\overline{D} + \overline{B}D + \overline{C}D + AB$$



Quine-McCluskeyeva metoda

- druga faza:
  - nakon nalaženja bitnih primarnih članova ~ moguća pojava *cikličke* tablice:
    - *Pyne-McCluskeyev pristup*: preostale primarne implikante tretirati kao logičke varijable i izgraditi funkciju F

      $F = (\text{suma pi koji prekrivaju } m_{i1}) \cdot$
      $\quad (\text{suma pi koji prekrivaju } m_{i2}) \cdot \ldots$
      $= \ldots = \text{suma produkata}$
    - uzeti produkt s minimalnim brojem primarnih implikanata

Quine-McCluskeyeva metoda

- minimizacija *nepotpuno specificiranih funkcija* u obliku sume produkata: $f = \sum_i m_i + \sum_j d_j$
  ~ modifikacija osnovnog postupka uvođenjem "vektora redundancija"
- postupak:
  - početna tablica
    ~ *mintermi* i *nespecificirane kombinacije*
  - svaki produktni član dobiva oznaku redundantnosti:
    d = 0 : produkt *nije* zanemariv
    ~ simplifikacija je uključila barem jedan m_i
    d = 1 : produkt je zanemariv ($\rightarrow$ redundancija!)
    ~ nastao kombiniranjem *samo* d_i

Quine-McCluskeyeva metoda

- prva faza:
  - kombiniranje produkata kao u osnovnom postupku, uz *evidenciju redundantnosti*
    $d = d_{i1} \cdot d_{i2}$: produkt zanemariv samo ako je nastao simplifikacijom zanemarivih produkata
  - *priprema* druge faze
    ~ izbor pi_i koji *nisu zanemarivi* ($d = 0$):
    - tablica za izbor minimalne sume
      ~ upis samo pi_i koji *nisu zanemarivi*
    - stupci tablice
      ~ m_i (X ne treba prekriti)
- druga faza postupka
  ~ identična osnovnom postupku

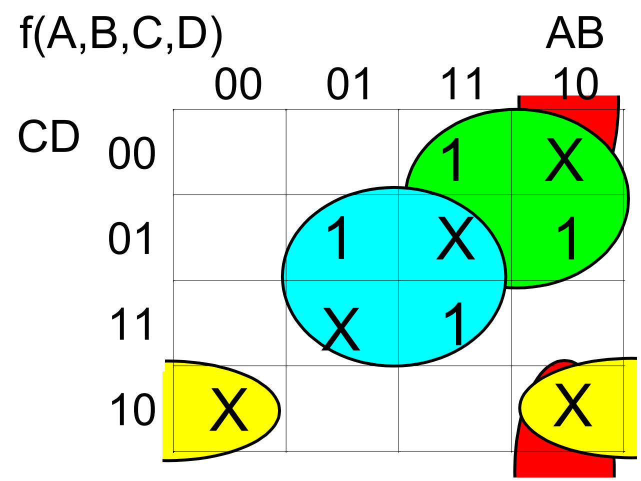
Quine-McCluskeyeva metoda

Primjer: $f(A,B,C,D)=\sum m(5,9,12,15)+\sum d(2,7,8,10,13)$

	ABCD	d	
2	0010	1	✓
8	1000	1	✓
5	0101	0	✓
9	1001	0	✓
10	1010	1	✓
12	1100	0	✓
7	0111	1	✓
13	1101	1	✓
15	1111	0	✓

	ABCD	d	
2,10	-010	1	
8,9	100-	0	✓
8,10	10-0	1	
8,12	1-00	0	✓
5,7	01-1	0	✓
5,13	-101	0	✓
9,13	1-01	0	✓
12,13	110-	0	✓
7,15	-111	0	✓
13,15	11-1	0	✓

	ABCD	d
8,9,12,13	1-0-	0
8,12,9,13	1-0-	0
5,7,13,15	-1-1	0
5,13,7,15	-1-1	0



potrebni pi_i (d=0): $\left\{A\overline{C}, BD\right\}$

nepotrebni pi_i (d=1): $\left\{\overline{B}C\overline{D}, A\overline{B}\,\overline{D}\right\}$

Quine-McCluskeyeva metoda

- druga faza :

		5	9	12	15
$A\overline{C}$	a		X	X	
BD	b	X			X
		✓	✓	✓	✓

$$f = A\overline{C} + BD$$

f(A,B,C,D)

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

Minimizacija višeizlazne funkcije

- *višeizlazna funkcija*
  ~ skup Booleovih funkcija nad
  istim skupom varijabli:
  definira "višeizlazni sklop"
  (engl. multiple-output circuit)

Primjer : pretvorba 3-bitovnog broja u (3-bitovni) Grayev kod

$$g_2 = \varphi_2(b_2, b_1, b_0)$$
$$g_1 = \varphi_1(b_2, b_1, b_0)$$
$$g_0 = \varphi_0(b_2, b_1, b_0)$$

b2	b1	b0	g2	g1	g0
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

Minimizacija višeizlazne funkcije

- minimizacija višeizlazne funkcije
  ~ mogućnosti:
  - zasebna minimizacija komponentnih funkcija f_i
  - *združena* minimizacija *svih* komponentnih funkcija višeizlazne funkcije (f_1, …, f_n)
    ~ povoljnije rješenje?
- minimizirana višeizlazna funkcija:
  - *višestruko* korištenje pojedinih produktnih članova
    ~ ušteda sklopovlja višeizlaznog sklopa
  - prilagodba (prethodnih) postupaka minimizacije
    ~ *istovremena* minimizacija komponentnih funkcija

Minimizacija višeizlazne funkcije

Primjer:

$$f_0 = AC + AB = pi_1 + pi_2,$$
$$= AC + AB\overline{C} = pi_1 + m_6$$

$f_0(A,B,C)$

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

$$f_1 = \overline{A}B + B\overline{C} = pi_3 + pi_4$$
$$= \overline{A}B + AB\overline{C} = pi_3 + m_6$$

$f_1(A,B,C)$

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

- višeizlazna funkcija $\{f_0, f_1\}$ ima povoljnije rješenje (pi_1, pi_3, m_6) u odnosu na zasebnu minimizaciju f_0 i f_1 što daje (pi_1, pi_2, pi_3, pi_4)

Minimizacija višeizlazne funkcije

- *konceptualizacija* postupka višeizlazne minimizacije:
  - *višeizlazni primarni implikant* pi_i *nije nužno* primarni implikant pojedinih funkcija:

$$pi_1, pi_2 \Rightarrow f_0$$
$$pi_3, pi_4 \Rightarrow f_1$$
$$m_6 = pi_5 \Rightarrow f_0 \cdot f_1$$

  - združena minimizacija n funkcija $f_1 \div f_n$:
    - odrediti pi_i $\forall$ f_i
    - odrediti pi_i $\forall$ kombinaciju f_i: produkti 2 i više f_i

Minimizacija višeizlazne funkcije

Primjer: $f(A, B, C, D) = \{f_1, f_2, f_3\}$

f_1

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

f_2

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

f_3

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

Minimizacija višeizlazne funkcije

Primjer: $f(A, B, C, D) = \{f_1, f_2, f_3\}$

f_1

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

f_2

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

f_3

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

$f_1 f_2$

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

$f_1 f_3$

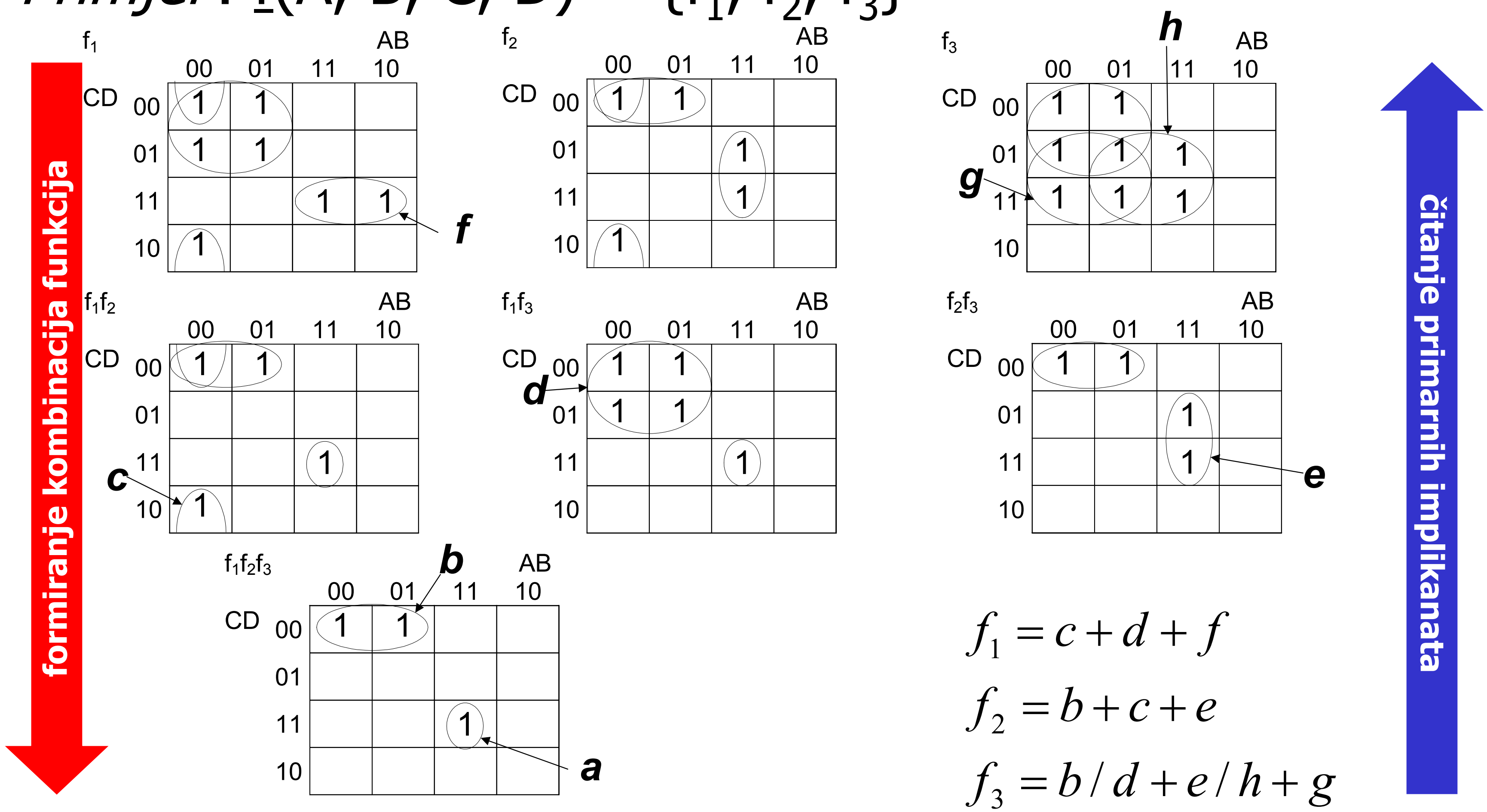
CD \ AB	00	01	11	10
00	1	1		
01	1	1		
11			1	
10				

$f_2 f_3$

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

Minimizacija višeizlazne funkcije

Primjer: $\underline{f}(A, B, C, D) = \{f_1, f_2, f_3\}$



$$f_1 = c + d + f$$

$$f_2 = b + c + e$$

$$f_3 = b / d + e / h + g$$

Minimizacija višeizlazne funkcije

- izbor *minimalnog skupa* višeizlaznih pi_i
  koji će prekrivati *sve tri* funkcije f_1, f_2, f_3:
  - povoljan izbor
    ~ pi_i koji se javljaju u max broju f_i:
    max zajedničko korištenje produkata
  - početi od $f_1 \cdot f_2 \cdot f_3$
  - izabrani složeniji pi_i javljaju se u "nižim" K tablicama kao zalihosti X
- komentar rješenja primjera:
  - *h* (f_3) ne doprinosi prekrivanju
  - f_2 ne daje p_i
  - *a* je nepotreban, jer ga prekrivaju *f*, *e*, *h*
  - f_3 ima opcije (*b* ili *d*, te *e* ili *h*)

Quine-McCluskey za višeizlazne funkcije

- sustavna metoda nalaženja minimalne sume
  ~ npr. modifikacija Quine-McCluskeyeve metode
  - prva faza
    ~ nalaženje skupa svih *višeizlaznih* primarnih implikanata (potpune sume)
    - prekrivanje komponentnih funkcija s pi_i
      ~ *vektor prekrivanja*
  - druga faza
    ~ nalaženje *minimalnog* skupa *višeizlaznih* primarnih implikanata (minimalne sume)

Quine-McCluskey za višeizlazne funkcije

- prva faza
  ~ utvrđivanje potpune sume:
  - raspodjela m_i *svih* f_i u indeksne grupe
    ~ broj 1
  - paziti na pripadnost m_i pojedinoj f_i
    ~ vektor prekrivanja!
  - označiti produkt (~ *nije* pi_i !) iz prethodne tablice jedino ako se u narednoj pojavljuje isti uzorak f_i, $\forall i$
  - $<f_1 f_2 \ldots f_n> = <00\ldots0>$ ("sve nule")
    nije valjani implikant

Quine-McCluskey za višeizlazne funkcije

Primjer : $f_1 = \sum(0,1,2,4,5,11,15), f_2 = \sum(0,2,4,13,15), f_3 = \sum(0,1,3,4,5,7,13,15)$

		$f_1f_2f_3$	
0	0000	111	✓
1	0001	101	✓
2	0010	110	✓
4	0100	111	✓
3	0011	001	✓
5	0101	101	✓
7	0111	001	✓
11	1011	100	✓
13	1101	011	✓
15	1111	111	

		$f_1f_2f_3$	
0,1	000-	101	✓
0,2	00-0	110	
0,4	0-00	111	
1,3	00-1	001	✓
1,5	0-01	101	✓
4,5	010-	101	✓
3,7	0-11	001	✓
5,7	01-1	001	✓
5,13	-101	001	✓
7,15	-111	001	✓
11,15	1-11	100	
13,15	11-1	011	

		$f_1f_2f_3$
0,1,4,5	0-0-	101
1,3,5,7	0-1	001
5,7,13,15	-1-1	001

a:	15	(111)	e:	13,15	(011)
c:	0, 2	(110)	d:	0, 1, 4, 5	(101)
b:	0, 4	(111)	g:	1, 3, 5, 7	(001)
f:	11,15	(100)	h:	5, 7,13,15	(001)

Quine-McCluskey za višeizlazne funkcije

- druga faza:

	f_1							f_2					f_3							
	0	1	2	4	5	11	15	0	2	4	13	15	0	1	3	4	5	7	13	15
a							x					x								x
b	x			x				x		x			x			x				
c	x		x					x	x											
d	x	x		x	x								x	x		x	x			
e											x	x							x	x
f						x	x													
g														x	x		x	x		
h																	x	x	x	x

  - rezultat:
    - bitni primarni implikanti: b, c, d, e, f, g
    - dobiveno je *potpuno prekrivanje*