FTN Novi Sad

Merni instrumenti - Digitalna elektronika

2. KOMBINACIONA LOGIKA

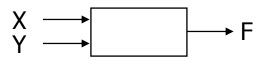
dr Zoran Mitrović

Kombinaciona logika

- Logičke funkcije, kombinacione tabele i prekidači
  - NE (NOT), I (AND), ILI (OR), NI (NAND), NILI (NOR), XOR, . . .
  - Minimalni skup
- Aksiome i teoreme bulove algebre
  - Provera pisanjem u drugom obliku
  - Provera indukcijom
- Logika gejtova
  - Mreže bulovih funkcija
  - Vremensko ponašanje
- Kanoničke forme
  - Dva nivoa
  - Nekompletno specificirane funkcije
- Uprošćenje
  - bulove kocke i karnoove mape
  - uprošćenje u dva nivoa

Moguće logičke funkcije dve promenljive

- Postoji 16 mogućih funkcija dve ulazne promenljive:
  - u opštem slučaju, postoji 2**(2**n) funkcija n ulaznih promenljivih



X	Y	16 mogućih funkcija (F0–F15)															
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		0	X and Y		X		Y	X xor Y	X or Y	X nor Y not (X or Y)	X = Y	not Y		not X		X nand Y not (X and Y)	1

"Cena" različitih logičkih funkcija

- Neke funkcije se lakše, a neke teže implementiraju
  - Svaka ima cenu koja zavisi od broja primenjenih prekidača
  - 0 (F0) i 1 (F15): zahtevaju 0 prekidača, direktno se vezuje izlaz na low/high
  - X (F3) i Y (F5): zahtevaju 0 prekidača, izlaz je jedan od ulaza
  - X' (F12) i Y' (F10): zahtevaju 2 prekidača za "invertor" ili NOT-gejt
  - X nor Y (F4) i X nand Y (F14): zahtevaju 4 prekidača
  - X or Y (F7) i X and Y (F1): zahtevaju 6 prekidača
  - X = Y (F9) i X ⊕ Y (F6): zahtevaju 16 prekidača

  - Pošto su NOT, NOR, i NAND najjeftiniji, to su funkcije koje se najčešće primenjuju u praksi

Minimalni skup funkcija

- Možemo li da implementiramo sve logičke funkcije koristeći NOT, NOR, i NAND?
  - Na primer, implementirati X and Y
    je isto kao implementirati not (X nand Y)
- **Ustvari, ovo može da se uradi samo sa NOR ili samo sa NAND**
  - NOT je NAND ili NOR sa oba ulaza spojena zajedno

X	Y	X nor Y
0	0	1
1	1	0

X	Y	X nand Y
0	0	1
1	1	0

  - i NAND i NOR su "duali", tj., jedno kolo se lako implementira pomoću drugog

X nand Y ≡ not ((not X) nor (not Y))
X nor Y ≡ not ((not X) nand (not Y))

- Pogledajmo prvo matematičke osnove logičkih kola...

Algebarske strukture

- Algebarska struktura sastoji se od
  - skupa elemenata B
  - binarnih operacija { + , • }
  - i unarne operacije { ' }
  - tako da važe sledeći aksiomi:

1. skup B sadrži najmanje dva elementa, a, b, takvih da je $a \neq b$
2. zatvaranje: a + b je u B — a • b je u B
3. komutativnost: a + b = b + a — a • b = b • a
4. asocijativnost: a + (b + c) = (a + b) + c — a • (b • c) = (a • b) • c
5. identičnost: a + 0 = a — a • 1 = a
6. distributivnost: a + (b • c) = (a + b) • (a + c) — a • (b + c) = (a • b) + (a • c)
7. komplementarnost: a + a' = 1 — a • a' = 0

Bulova algebra

- Bulova algebra
  - B = {0, 1}
  - + je logičko OR, • je logičko AND
  - ' je logičko NOT
- Sve algebarske aksiome važe

Logičke funkcije i bulova algebra

- Svaka logička funkcija koja može da se predstavi kao kombinaciona tabela može da se napiše kao izraz u bulovoj algebri korišćenjem operatora: ', +, and •

X	Y	X • Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X'	X' • Y
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0

X	Y	X'	Y'	X • Y	X' • Y'	(X • Y) + (X' • Y')
0	0	1	1	0	1	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	1	0	1

(X • Y) + (X' • Y') ▯ X = Y

Bulov izraz koji je tačan kad promenljive X i Y imaju istu vrednost, u suprotnom netačan

X, Y su promenljive bulove algebre

Aksiome i teoreme bulove algebre

- Identičnost
  1. $X + 0 = X$ — 1D. $X \cdot 1 = X$
- Nuliranje
  2. $X + 1 = 1$ — 2D. $X \cdot 0 = 0$
- Idempotencija:
  3. $X + X = X$ — 3D. $X \cdot X = X$
- Involucija:
  4. $(X')' = X$
- Komplementarnost:
  5. $X + X' = 1$ — 5D. $X \cdot X' = 0$
- Komutativnost:
  6. $X + Y = Y + X$ — 6D. $X \cdot Y = Y \cdot X$
- Asocijativnost:
  7. $(X + Y) + Z = X + (Y + Z)$ — 7D. $(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$

Aksiome i teoreme bulove algebre (nastavak)

- Distributivnost:
  8. $X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$ — 8D. $X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$
- Unarna operacija:
  9. $X \cdot Y + X \cdot Y' = X$ — 9D. $(X + Y) \cdot (X + Y') = X$
- Apsorpcija:
  10. $X + X \cdot Y = X$ — 10D. $X \cdot (X + Y) = X$
  11. $(X + Y') \cdot Y = X \cdot Y$ — 11D. $(X \cdot Y') + Y = X + Y$
- Faktorizacija:
  12. $(X + Y) \cdot (X' + Z) = X \cdot Z + X' \cdot Y$ — 12D. $X \cdot Y + X' \cdot Z = (X + Z) \cdot (X' + Y)$
- Konsenzus:
  13. $(X \cdot Y) + (Y \cdot Z) + (X' \cdot Z) = X \cdot Y + X' \cdot Z$ — 17D. $(X + Y) \cdot (Y + Z) \cdot (X' + Z) = (X + Y) \cdot (X' + Z)$

Aksiome i teoreme bulove algebre (nastavak)

- de Morganova pravila:
  - 14. (X + Y + ...)' = X' • Y' • ...   14D. (X • Y • ...)' = X' + Y' + ...
- generalizovana de Morganova pravila:
  - 15. f'(X1,X2,...,Xn,0,1,+,•) = f(X1',X2',...,Xn',1,0,•,+)
- uspostavljaju vezu između • and +

Aksiome i teoreme bulove algebre (nastavak)

- Dualnost
  - Dualni izraz bulovom izrazu dobija se zamenom • sa +, + sa •, 0 sa 1, i 1 sa 0, a promenljive se ostavljaju kakve su bile
  - Bilo koja teorema koja može da se dokaže važi takođe i za dualnu funkciju
  - Meta-teorema (teorema o teoremama)
- dualnost:
  - 16. X + Y + ... ⇔ X • Y • ...
- generalizovana dualnost:
  - 17. f (X1,X2,...,Xn,0,1,+,•) ⇔ f(X1,X2,...,Xn,1,0,•,+)
- Različito od deMorganovog zakona (pravila)
  - ovo je izraz o teoremama
  - to nije način da se drugačije napišu izrazi

Provera (dokaz) teorema (drugačijim pisanjem)

- Korišćenjem aksioma bulove algebre:
  - npr., dokaži teoremu: $X \cdot Y + X \cdot Y' = X$

distributivnost (8)	$X \cdot Y + X \cdot Y'$	=	$X \cdot (Y + Y')$
komplementarnost (5)	$X \cdot (Y + Y')$	=	$X \cdot (1)$
identičnost (1D)	$X \cdot (1)$	=	X ✓

  - npr., dokaži teoremu: $X + X \cdot Y = X$

identičnost (1D)	$X + X \cdot Y$	=	$X \cdot 1 + X \cdot Y$
distributivnost (8)	$X \cdot 1 + X \cdot Y$	=	$X \cdot (1 + Y)$
identičnost (2)	$X \cdot (1 + Y)$	=	$X \cdot (1)$
identičnost (1D)	$X \cdot (1)$	=	X ✓

Provera (dokaz) teorema (indukcijom)

- Korišćenjem potpune indukcije (kompletne kombinacione tabele):
  - npr., de Morganova pravila:

$(X + Y)' = X' \cdot Y'$
NOR je ekvivalentno sa AND uz komplementiranje ulaza

X	Y	X'	Y'	(X + Y)'	X' · Y'
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

$(X \cdot Y)' = X' + Y'$
NAND je ekvivalentno sa OR uz komplementiranje ulaza

X	Y	X'	Y'	(X · Y)'	X' + Y'
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

Jednostavan primer

- 1-bitni binarni sabirač
  - ulazi: A, B, Carry-in
  - izlazi: Zbir, Carry-out

A	B	Cin	Zbir	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Zbir = A' B' Cin + A' B Cin' + A B' Cin' + A B Cin

Cout = A' B Cin + A B' Cin + A B Cin' + A B Cin

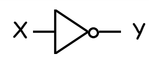
Primena teorema za uprošćenje izraza

- Teoreme bulove algebre mogu da uproste bulove izraze
  - npr., jednačine za izlaz prenosa (carry-out) punog sabirača (ista pravila važe za bilo koju funkciju)

Cout = A' B Cin + A B' Cin + A B Cin' + A B Cin
= A' B Cin + A B' Cin + A B Cin' + A B Cin + A B Cin
= A' B Cin + A B Cin + A B' Cin + A B Cin' + A B Cin
= (A' + A) B Cin + A B' Cin + A B Cin' + A B Cin
= (1) B Cin + A B' Cin + A B Cin' + A B Cin
= B Cin + A B' Cin + A B Cin' + A B Cin + A B Cin
= B Cin + A B' Cin + A B Cin + A B Cin' + A B Cin
= B Cin + A (B' + B) Cin + A B Cin' + A B Cin
= B Cin + A (1) Cin + A B Cin' + A B Cin
= B Cin + A Cin + A B (Cin' + Cin)
= B Cin + A Cin + A B (1)
= B Cin + A Cin + A B

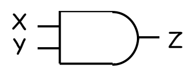
Od bulovih izraza do logičkih gejtova

- NOT X' $\overline{X}$ $\sim X$



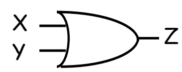
X	Y
0	1
1	0

- AND $X \cdot Y$ XY $X \wedge Y$



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

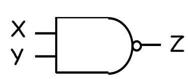
- OR $X + Y$ $X \vee Y$



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

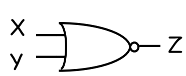
Od bulovih izraza do logičkih gejtova (nastavak)

- NAND



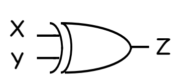
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

- NOR



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

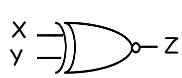
- XOR
  $X \oplus Y$



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

X xor Y = X Y' + X' Y
X ili Y ali ne oba
("nejednakost", "razlika")

- XNOR
  $X = Y$

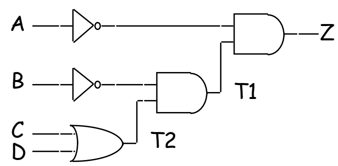


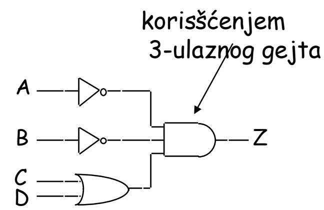
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

X xnor Y = X Y + X' Y'
X i Y su isti
("jednakost", "koincidencija"

Od bulovih izraza do logičkih gejtova (nastavak)

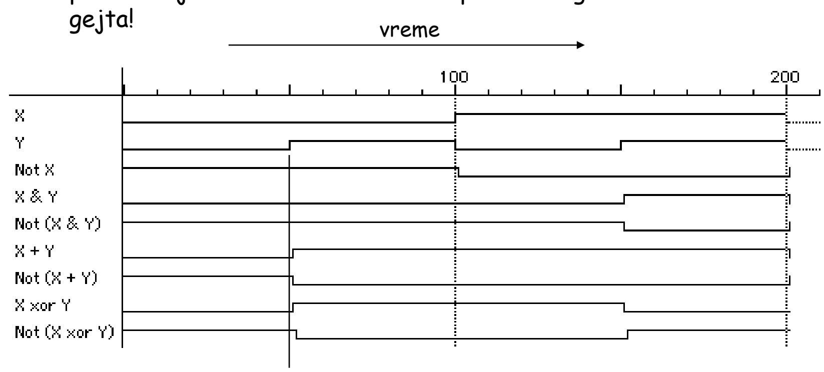
- Više od jednog načina da se izraz mapira u gejtove
  - npr., $Z = A' \cdot B' \cdot (C + D) = (A' \cdot \underbrace{(B' \cdot \overbrace{(C + D)}^{T2})}_{T1})$





Talasni oblici logičkih funkcija

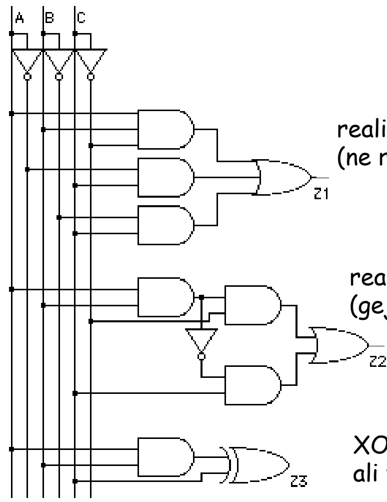
- praktično je to rotirana kombinaciona tabela
  - Uočiti kako se ivice ne nalaze na istoj liniji
  - potrebno je određeno vreme da se prenese signal sa ulaza na izlaz gejta!



promena Y se nakon određenog vremena "propagira" kroz gejtove

Izbor različitih realizacija funkcije

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



realizacija u dva nivoa
(ne računamo NOT gejtove)

realizacija u više nivoa
(gejtovi sa manje ulaza)

XOR gejt (lakši za crtanje,
ali više košta da se napravi)

Koja realizacija je najbolja?

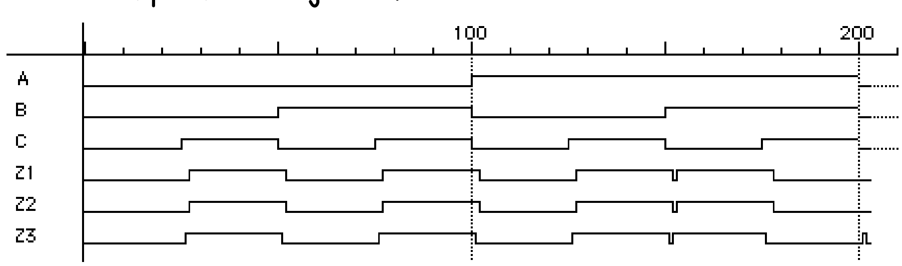
- Smanjiti broj ulaza
  - ulazna promenljiva (komplementirana ili ne)
    - može približno da košta kao dva tranzistora po ulazu
    - Zašto se ne računaju invertori?
  - Manje ulaza znači manje tranzistora
    - manja kola
  - Manje ulaza pretpostavlja brže gejtova
    - gejtovi su manji i zbog toga i brži
  - Fan-in (broj ulaza gejta) je ograničen u nekim tehnologijama
- Smanjiti broj gejtova
  - Manje gejtova (i pakovanja u kojima se isporučuju) znači manja elektronska kola
    - direktno utiče na troškove proizvođača

Koja realizacija je najbolja? (nastavak)

- Smanjiti broj nivoa gejtova
  - Manji broj nivoa gejtova znači manju propagaciju (kašnjenje)
  - Konfiguracija sa manjim kašnjenjem zahteva više gejtova
    - šira kola sa manjom dubinom
- Kako koristimo "trgovinu" između povećanog kašnjenja i veličine kola?
  - Automatizovani alati za generisanje različitih rešenja
  - Logička minimizacija: smanjiti broj gejtova i kompleksnost
  - Logička optimizacija: redukcija uz uzimanje u obzir kašnjenja

Jesu li sve realizacije ekvivalentne?

- Uz iste ulazne stimuluse, tri alternativne implementacije imaju skoro isto ponašanje
  - kašnjenja su različita
  - gličevi mogu da se pojave
  - varijacije zbog razlika u broju nivoa gejtova i strukturi
- Tri implementacije su funkcionalno ekvivalentne



Implementiranje bulovih funkcija

- Tehnološki nezavisne
  - Kanoničke forme
  - Forme sa dva nivoa
  - Forme sa više nivoa
- Izbor tehnologije
  - Pakovanje sa nekoliko gejtova
  - Regularna logička kola
  - Programabilna logika sa dva nivoa
  - Programabilna logika sa više nivoa

Kanoničke forme

- Kombinacione tabele su jedinstveno predstavljanje bulovih funkcija
- Mnogo alternativnih realizacija gejtova mogu da imaju istu kombinacionu tabelu
- Kanoničke forme
  - Standardne forme za bulove izraze
  - Obezbeđuje jedinstveni algebarski “potpis”

Zbir-proizvoda (Sum-of-Products) kanoničke forme

- Takođe poznate kao disjunktne normalne forme
- Takođe poznate kao minterm proširenje

F = *001* *011* *101* *110* *111*

F = A'B'C+ A'BC+ AB'C + ABC' + ABC

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

F' = A'B'C' + A'BC' + AB'C'

Zbir-proizvoda (Sum-of-Products) kanoničke forme (nastavak)

- Product term (ili minterm)
  - ANDovan proizvod činilaca – ulazna kombinacija za koju je izlaz TRUE
  - Svaka promenljiva pojavljuje se samo jednom, u invertovanoj ili neinvertovanoj formi (ali ne u obe)

A	B	C	minterm-ovi
0	0	0	A'B'C' m0
0	0	1	A'B'C m1
0	1	0	A'BC' m2
0	1	1	A'BC m3
1	0	0	AB'C' m4
1	0	1	AB'C m5
1	1	0	ABC' m6
1	1	1	ABC m7

skraćena notacija za minterms 3 promenljive

F u kanoničkoj formi:

F(A, B, C) = Σm(1,3,5,6,7)
= m1 + m3 + m5 + m6 + m7
= A'B'C + A'BC + AB'C + ABC' + ABC

kanonička forma $\neq$ minimalna forma

F(A, B, C) = A'B'C + A'BC + AB'C + ABC + ABC'
= (A'B' + A'B + AB' + AB)C + ABC'
= ((A' + A)(B' + B))C + ABC'
= C + ABC'
= ABC' + C
= AB + C

Proizvod-zbirova (Product-of-Sums) kanonička forma

- Poznata i kao konjunktivna normalna forma
- Poznata i kao maxterm ekspanzija

F = *000* *010* *100*

F = (A + B + C) (A + B' + C) (A' + B + C)

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

F' = (A + B + C') (A + B' + C') (A' + B + C') (A' + B' + C) (A' + B' + C')

Proizvod-zbirova (Product-of-Sums) kanonička forma (nastavak)

- Sum term (ili maxterm)
  - ORovana suma činilaca – ulazna kombinacija za koju je izlaz FALSE
  - Svaka promenljiva pojavljuje se samo jednom, u invertovanoj ili neinvertovanoj formi (ali ne u obe)

A	B	C	maxterm-ovi	
0	0	0	A+B+C	M0
0	0	1	A+B+C'	M1
0	1	0	A+B'+C	M2
0	1	1	A+B'+C'	M3
1	0	0	A'+B+C	M4
1	0	1	A'+B+C'	M5
1	1	0	A'+B'+C	M6
1	1	1	A'+B'+C'	M7

skraćena notacija za maxterm-ove 3 promenljive

F u kanoničkoj formi:

F(A, B, C) = ΠM(0,2,4)
= M0 · M2 · M4
= (A + B + C) (A + B' + C) (A' + B + C)

kanonička forma $\neq$ minimalna forma

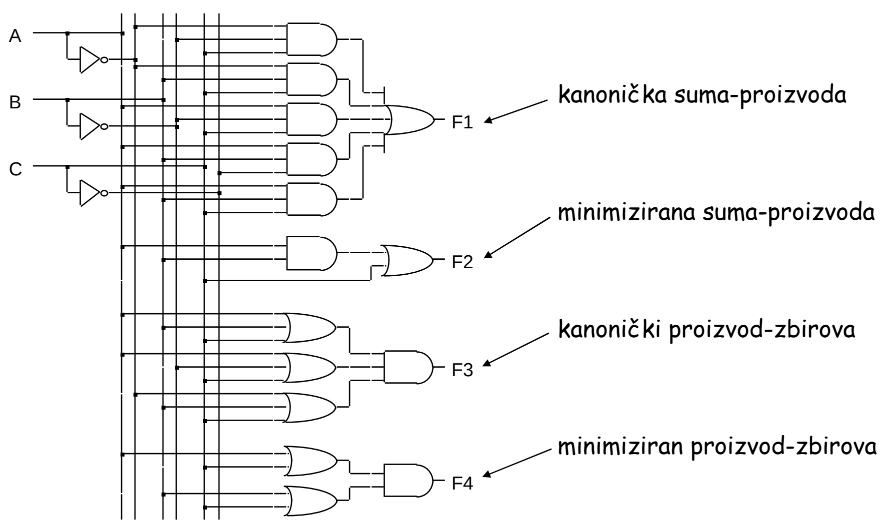
F(A, B, C) = (A + B + C) (A + B' + C) (A' + B + C)
= (A + B + C) (A + B' + C)
(A + B + C) (A' + B + C)
= (A + C) (B + C)

S-o-P, P-o-S, i de Morganova teorema

- Sum-of-products (zbir-proizvoda)
  - F' = A'B'C' + A'BC' + AB'C'
- Primenimo de Morganovu teoremu
  - (F')' = (A'B'C' + A'BC' + AB'C')'
  - F = (A + B + C) (A + B' + C) (A' + B + C)

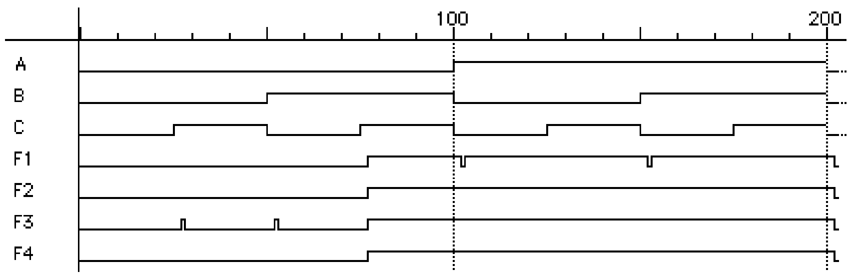
- Product-of-sums (proizvod-zbirova)
  - F' = (A + B + C') (A + B' + C') (A' + B + C') (A' + B' + C) (A' + B' + C')
- Primenimo de Morganovu teoremu
  - (F')' = ((A + B + C')(A + B' + C')(A' + B + C')(A' + B' + C)(A' + B' + C'))'
  - F = A'B'C + A'BC + AB'C + ABC' + ABC

Četiri alternative za implementaciju funkcije F = AB + C u dva nivoa



Talasni oblici za četiri alternative

- Talasni oblici su u suštini identični
  - Izuzev gličeva
  - Kašnjenja su skoro identična (ako se modeliraju kao kašnjenje po nivou, na po tipu gejta ili po broju ulaza gejta)



Mapiranje između četiri kanoničke forme

- Konverzija minterm u Maxterm
  - Koristiti Maxterm-ove čiji se indeksi ne pojavljuju u minterm ekspanziji
  - npr., $F(A,B,C) = \Sigma m(1,3,5,6,7) = \Pi M(0,2,4)$
- Konverzija Maxterm u minterm
  - Koristiti minterm-ove čiji indeksi se ne pojavljuju u Maxterm ekspanziji
  - npr., $F(A,B,C) = \Pi M(0,2,4) = \Sigma m(1,3,5,6,7)$
- Minterm ekspanzija F u minterm ekspanziju F'
  - Koristiti minterm-ove čiji indeksi se ne pojavljuju
  - npr., $F(A,B,C) = \Sigma m(1,3,5,6,7)$ $F'(A,B,C) = \Sigma m(0,2,4)$
- Maxterm ekspanzija F u maxterm ekspanziju F'
  - Koristiti minterm-ove čiji indeksi se ne pojavljuju
  - npr., $F(A,B,C) = \Pi M(0,2,4)$ $F'(A,B,C) = \Pi M(1,3,5,6,7)$

Nepotpuno specificirane funkcije

- Primer: inkrementiranje za 1 binarno kodiranog decimalnog broja
  - BCD cifre predstavljaju decimalne cifre 0 – 9 preko četvorobitnih sekvenci 0000 – 1001

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
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	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

isključena stanja W

uključena stanja W

nije-važno
(don't care, DC)
stanja W

ova ulazna stanja ne bi trebalo da se sretnu u praksi– "don't care" stanja mogu da se iskoriste u minimizaciji

Notacija za nekompletno specificirane funkcije

- Stanja nije-važno (don't care) i kanoničke forme
  - Do sada, jedino su se prikazivala stanja uključeno (TRUE)
  - Takođe se predstavlja skup stanja don't-care
  - Potrebna su nam dva od tri skupa (stanja uključeno, stanja isključeno, stanja nije-važno)
- Kanoničko predstavljanje funkcije inkrementiranja za 1 BCD broja:
  - Z = m0 + m2 + m4 + m6 + m8 + d10 + d11 + d12 + d13 + d14 + d15
  - Z = Σ [m(0,2,4,6,8) + d(10,11,12,13,14,15)]
  - Z = M1 • M3 • M5 • M7 • M9 • D10 • D11 • D12 • D13 • D14 • D15
  - Z = Π [M(1,3,5,7,9) • D(10,11,12,13,14,15)]

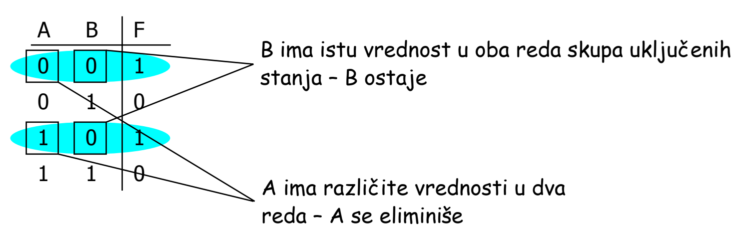
Uprošćavanje kombinacione logike u dva nivoa

- Pronalaženje realizacija minimalne sume proizvoda ili proizvoda zbirova (suma)
  - Iskoristiti informaciju o stanjima nije-važno u ovom procesu
- Algebarsko uprošćenje
  - Nije algoritamsko/sistematska procedura
  - Kako znamo kada smo našli minimalnu realizaciju?
- Računarski razvojni alati
  - Precizna rešenja zahtevaju velika računska vremena, posebno za funkcije sa mnogo ulaza (> 10)
  - Heurističke metode se primenjuju – "naučeni pogoci" da bi se smanjio broj računanja i da bi se došlo do dobrog, ako ne i najboljeg rezultata
- "Ručne" metode su još relevantne
  - Da bismo razumeli automatizovane alate i njihovu snagu i slabosti
  - Mogućnost provere rezultata (na malim primerima)

Teorema sjedinjenja

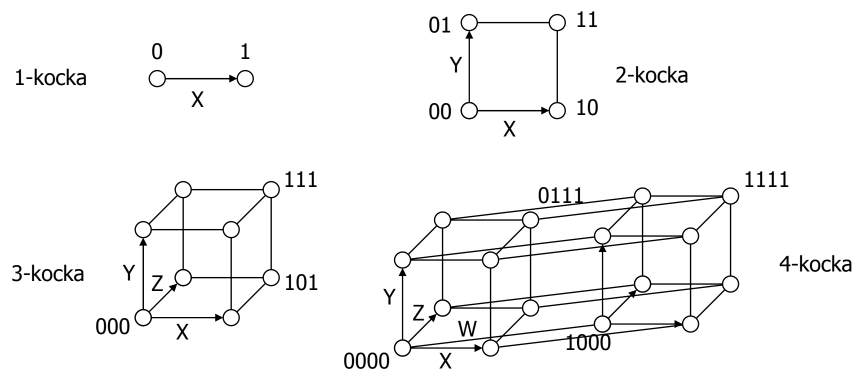
- Ključno sredstvo uprošćenja: A (B' + B) = A
- Suština uprošćenja logike u dva nivoa
  - Pronaći dva podskupa elemenata iz skupa uključenih stanja gde samo jedna promenljiva menja svoju vrednost – ova jedina promenljiva vrednost može da se eliminiše i da se jedan product term koristi za predstavljanje oba elementa

$$F = A'B' + AB' = (A' + A)B' = B'$$



Bulove kocke

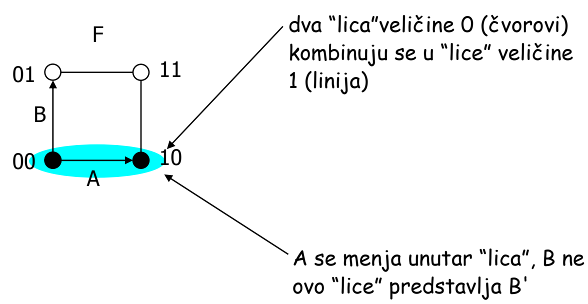
- ⌘ Vizuelna tehnika za određivanje kad može da se primeni teorema sjedinjenja
- ⌘ n ulaznih promenljivih = n-dimenzionalna “kocka”



Mapiranje kombinacionih tabela u bulove kocke

- ⌘ Teorema sjedinjenja kombinuje dva “lica” kocke u veće “lice”
- ⌘ Primer:

A	B	F
0	0	1
0	1	0
1	0	1
1	1	0

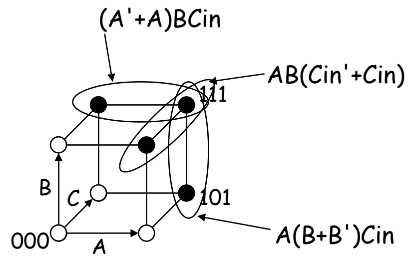


skup uključenih stanja = popunjeni čvorovi
skup isključenih stanja = prazni čvorovi
skup stanja nije-važno = čvorovi označeni sa ×

Primer sa tri promenljive

⌘ Binarni pun sabirač – logika za signal prenosa (carry)

A	B	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Skup uključenih stanja je potpuno pokrivena kombinacijom (OR) podkocki niže dimenzije – primetiti da je "111" pokriveno tri puta

Cout = BCin+AB+ACin

Kocke sa više dimenzija

⌘ Pod-kocke dimenzije veće od 2

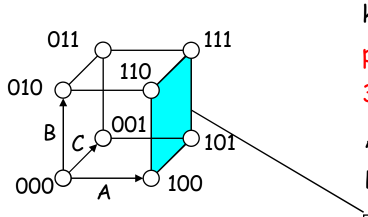
F(A,B,C) = Σm(4,5,6,7)

skup uključenih stanja formira kvadrat, tj. kocku dimenzije 2

predstavlja prikaz u jednoj promenljivoj, tj. 3 dimenzije – 2 dimenzije

A je TRUE i nepromenjeno

B i C se menjaju



Ova podkocka predstavlja promenljivu A

m-dimenzione kocke u n-dimenzionom bulovom prostoru

- U 3-kocki (tri promenljive):
  - 0-kocka, tj., jedinični čvor, izraz sa 3 promenljive
  - 1-kocka, tj., linija sa dva čvora, izraz sa 2 promenljive
  - 2-kocka, tj., ravan sa četiri čvora, izraz sa 1 promenljivom
  - 3-kocka, tj., kocka sa osam čvorova, izraz sa konstantom "1"
- U opštem slučaju,
  - m-podkocka unutar n-kocke (m < n) – izraz sa n – m promenljivih

Karnoove mape

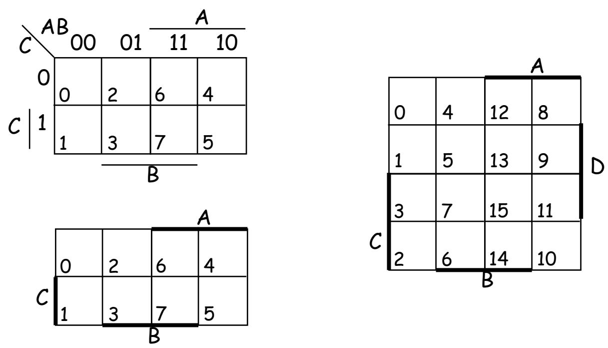
- Predstava bulove kocke u ravni
  - Savija se na ivicama
  - Teška za crtanje i vizualizaciju za više od 4 dimenzije
  - Praktično nemoguće za više od 6 dimenzija
- Alternativa kombinacionim tabelama za pomoć vizualizaciji susednih stanja
  - Vodič za primenu teoreme sjedinjenja
  - Elementi skupa uključenog stanja sa samo jednom promenljivom koja menja vrednost su susedna, suprotno situaciji u linearnoj kombinacionoj tabeli

B \ A	0	1
0	1 [0]	1 [2]
1	0 [1]	0 [3]

A	B	F
0	0	1
0	1	0
1	0	1
1	1	0

Karnoove mape (nastavak)

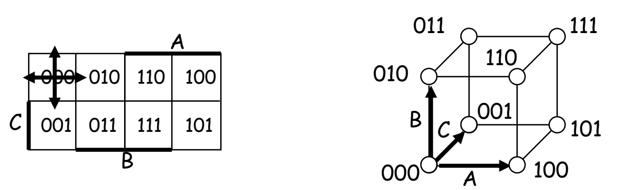
- Šema označavanja bazirana na grejovom kodu
  - npr., 00, 01, 11, 10
  - Samo jedan bit se menja između susednih ćelija



13 = 1101= ABC'D

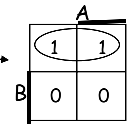
Susedne ćelije u Karnoovim mapama

- Prva i poslednja kolona su "susedne"
- Gornji i donji red su "susedni"



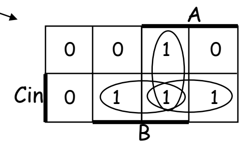
Primeri karnoovih mapa

- F =



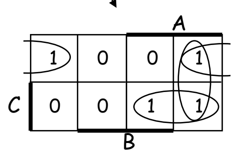
B'

- Cout =



AB + ACin+ BCin

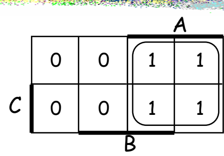
- $f(A,B,C) = \Sigma m(0,4,6,7)$



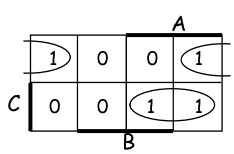
AC+ B'C' + ~~AB'~~

pronaći komplement funkcije pokrivanjem stanja 0 podkockama

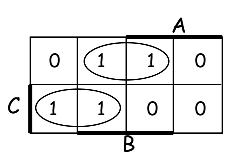
Primeri karnoovih mapa (nastavak)



G(A,B,C) = A



$F(A,B,C) = \Sigma m(0,4,5,7) = AC + B'C'$



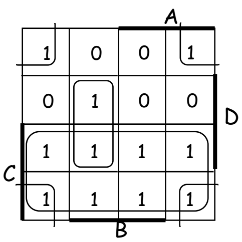
F' prosto zamenjuje jedinice nulama i obratno

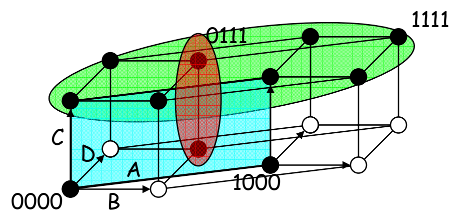
$F'(A,B,C) = \Sigma\ m(1,2,3,6) = BC' + A'C$

Karnoova mapa: primer sa 4 promenljive

- $F(A,B,C,D) = \Sigma m(0,2,3,5,6,7,8,10,11,14,15)$

  $F = C + A'BD + B'D'$

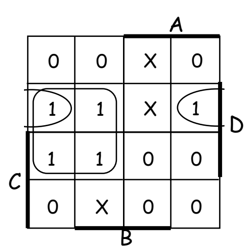




naći najmanji broj najvećih mogućih podkocki
da se pokrije skup uključenih stanja
(manje izraza sa manjim brojem ulaza po izrazu)

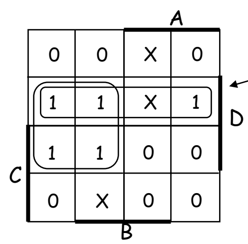
Karnoove mape: stanja nije-važno (don't care)

- $f(A,B,C,D) = \Sigma\, m(1,3,5,7,9) + d(6,12,13)$
  - bez stanja nije-važno
    - $f = A'D + B'C'D$



Karnoove mape: stanja nije-važno (don't care) (nastavak)

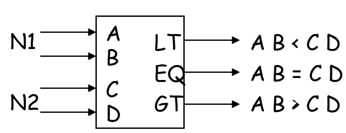
- $f(A,B,C,D) = \Sigma\, m(1,3,5,7,9) + d(6,12,13)$
  - $f = A'D + B'C'D$ — bez stanja nije-važno
  - $f = A'D + C'D$ — sa stanjima nije-važno



ako se stanje nije-važno koristi kao "1", 2-kocka može da se formira umesto 1-kocke da se pokrije ovaj čvor

<u>stanja nije-važno</u> mogu da se koriste kao 1 ili 0 zavisno od toga šta nam više odgovara

Primer: dvobitni komparator

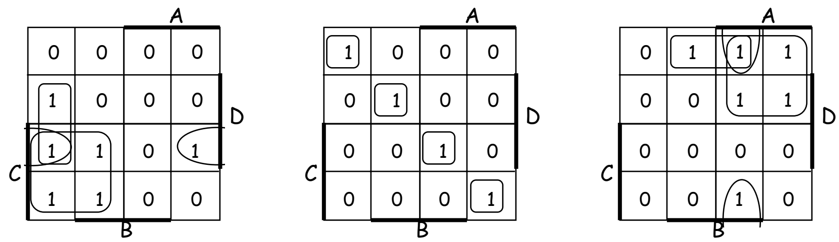


blok dijagram
i kombinaciona tabela

A	B	C	D	LT	EQ	GT
0	0	0	0	0	1	0
		0	1	1	0	0
		1	0	1	0	0
		1	1	1	0	0
0	1	0	0	0	0	1
		0	1	0	1	0
		1	0	1	0	0
		1	1	1	0	0
1	0	0	0	0	0	1
		0	1	0	0	1
		1	0	0	1	0
		1	1	1	0	0
1	1	0	0	0	0	1
		0	1	0	0	1
		1	0	0	0	1
		1	1	0	1	0

treba nam karnoova mapa sa 4 promenljive
za svaku od 3 izlazne funkcije

Primer: dvobitni komparator (nastavak)



K-mapa za LT　　K-mapa za EQ　　K-mapa za GT

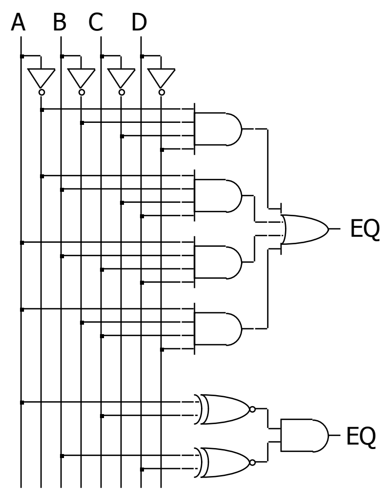
LT = A' B' D + A' C + B' C D

EQ = A'B'C'D' + A'BC'D + ABCD + AB'CD' = (A xnor C) • (B xnor D)

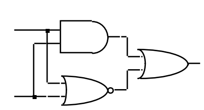
GT = B C' D' + A C' + A B D'

LT i GT su slični (zameniti A/C i B/D)

Primer: dvobitni komparator (nastavak)

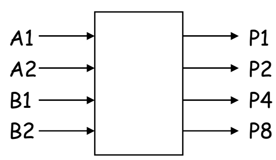


dve alternativne implementacije funkcije EQ sa i bez XOR



XNOR se implementira sa najmanje 3 prosta gejta

Primer: 2x2-bitni množač



blok dijagram
i kombinaciona tabela

A2	A1	B2	B1	P8	P4	P2	P1
0	0	0	0	0	0	0	0
		0	1	0	0	0	0
		1	0	0	0	0	0
		1	1	0	0	0	0
0	1	0	0	0	0	0	0
		0	1	0	0	0	1
		1	0	0	0	1	0
		1	1	0	0	1	1
1	0	0	0	0	0	0	0
		0	1	0	0	1	0
		1	0	0	1	0	0
		1	1	0	1	1	0
1	1	0	0	0	0	0	0
		0	1	0	0	1	1
		1	0	0	1	1	0
		1	1	1	0	0	1

K-mapa sa 4 promenljive
za svaku od 4
izlazne funkcije

Primer: 2x2-bitni množač (nastavak)

K-mapa za P8

			A2	A2	
	0	0	0	0	
	0	0	0	0	B1
B2	0	0	1	0	B1
B2	0	0	0	0	
		A1	A1		

P8 = A2A1B2B1

K-mapa za P4

			A2	A2	
	0	0	0	0	
	0	0	0	0	B1
B2	0	0	0	1	B1
B2	0	0	1	1	
		A1	A1		

P4 = A2B2B1'
+ A2A1'B2

K-mapa za P2

			A2	A2	
	0	0	0	0	
	0	0	1	1	B1
B2	0	1	0	1	B1
B2	0	1	1	0	
		A1	A1		

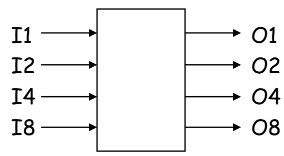
P2 = A2'A1B2
+ A1B2B1'
+ A2B2'B1
+ A2A1'B1

K-mapa za P1

			A2	A2	
	0	0	0	0	
	0	1	1	0	B1
B2	0	1	1	0	B1
B2	0	0	0	0	
		A1	A1		

P1 = A1B1

Primer: BCD inkrement za 1



blok dijagram
i kombinaciona tabela

I8	I4	I2	I1	O8	O4	O2	O1
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
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	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

K-mapa sa 4 promenljive za svaku
od 4 izlazne funkcije

Primer: BCD inkrement za 1 (nastavak)

O8 (I8, I1, I2, I4)

0	0	X	1
0	0	X	0
0	1	X	X
0	0	X	X

O4 (I8, I1, I2, I4)

0	1	X	0
0	1	X	0
1	0	X	X
0	1	X	X

O8 = I4 I2 I1 + I8 I1'
O4 = I4 I2' + I4 I1' + I4' I2 I1
O2 = I8' I2' I1 + I2 I1'
O1 = I1'

O2 (I8, I1, I2, I4)

0	0	X	0
1	1	X	0
0	0	X	X
1	1	X	X

O1 (I8, I1, I2, I4)

1	1	X	1
0	0	X	0
0	0	X	X
1	1	X	X