

Sinteza kombinacionih mreža



Katedra za elektroniku
prof dr Lazar Saranovac

Digitalna elektronika 1 - 2021/22

1

1

Kombinacione mreže su digitalni sistemi kod kojih izlazi zavise samo od stanja ulaza. ???
Od kojih stanja ulaza? Jučerašnjih? Od onih koja su bila pre 100ms? Bivših, sadašnjih, budućih?
Moramo dodati vremensku odrednicu!

Kombinacione mreže su digitalni sistemi kod kojih izlazi zavise samo od **trenutnog** stanja ulaza.

Trenutnog? Pojam koji često koristimo ne ulazeći dublje u njegovo značenje.

Ako su se ulazi promenili u nekom trenutku vremena da li su se i izlazi **trenutno** promenili.

Ako kažemo da se nešto trenutno dešava obično podrazumevamo neke vremenske margine pošto u realnom fizičkom svetu ne postoji pojava, promena, koja se zaista dešava trenutno, odnosno u beskonačno kratkom vremenskom intervalu.

Šta može da se desi i šta se dešava u digitalnom sistemu realizovanom kao kombinaciona mreža?

Ako se u nekom trenutku promene ulazi neće se odmah promeniti i izlazi. Postoji vremenski interval u kojem gornja tvrdnja ne važi. Stanja izlaza nam nisu poznata u tom vremenskom intervalu.

Zašto?

Logička kola ne mogu trenutno da „reaguju“ na promenu ulaznih promenljivih. Fizika. Izlazi će se promeniti tek posle nekog vremena. **Kašnjenje logičkog kola.**



Katedra za elektroniku
prof dr Lazar Saranovac

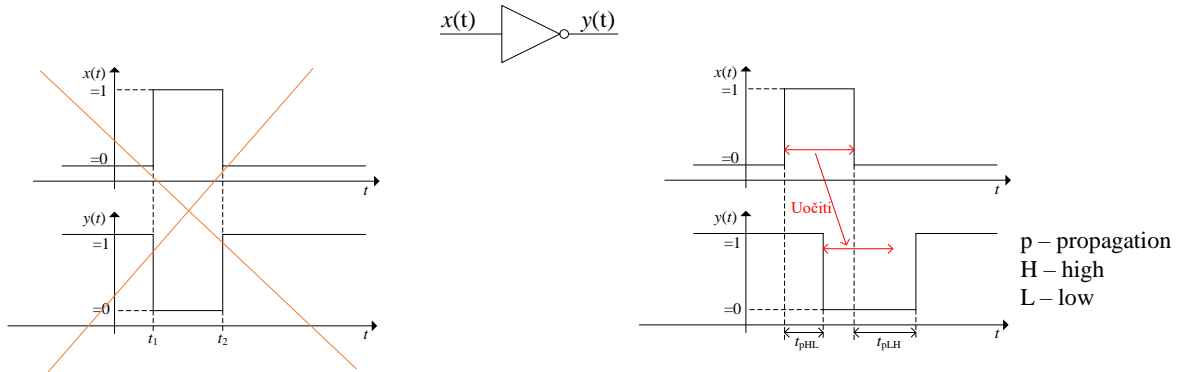
Digitalna elektronika 1 - 2021/22

2

2

Ulazni signal u logičko kolo se menja u vremenu

Signal – fizička veličina koja nosi informaciju o logičkim jedinicama i nulama.
 Za sada još uvek razmišljamo na apstraktnom nivou sa logičkim jedincama i nulama



t_{pHL} - kašnjenje izlaznog signala u odnosu na promenu ulaznog signala prilikom prelaska IZLAZNOG signala sa „nivoa“ H na L
 t_{pLH} - kašnjenje izlaznog signala u odnosu na promenu ulaznog signala prilikom prelaska IZLAZNOG signala sa „nivoa“ L na H
 H = 1, L = 0



3

Zapamtiti

$$t_{pHL} \neq t_{pLH}$$

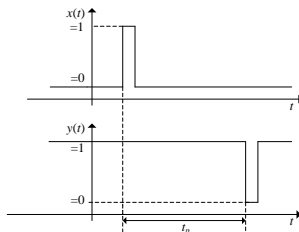
Uvešćemo pojam i srednjeg kašnjenja = kašnjenje logičkog kola

$$t_p = \frac{t_{pHL} + t_{pLH}}{2}$$

i radi jednostavnosti posmatranja **nekih** pojava smatrati

$$t_{pHL} = t_{pLH} = t_p$$

A onda često pogrešno razmišljanje za **sve** slučajeve



Signal na izlazu će biti isti samo zakašnjen?

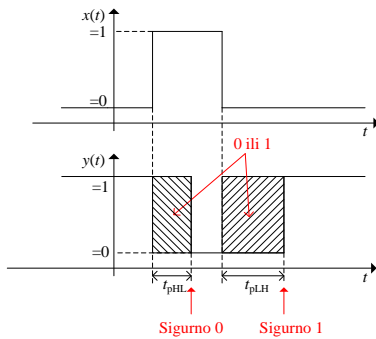
Dolazi do promene izlaznog signala, trajanja, a moguće da se neće ni pojaviti!

Zavisi od
 Trajanje signala PREMA kašnjenjima



4

Šta su u stvari inženjerski vremena t_{pHL} i t_{pLH}



Promena izlaznog signala može da se desi i ranije ali se SIGURNO dešava posle vremena t_{pHL} odnosno t_{pLH}

Znači ovo su MAKSIMALNA vremena posle kojih smo sigurni u stanje izlaznog signala

Stvarno kašnjenje logičkog kola će zavisti od napajanja, temperature, starosti komponente, ... ali neće biti veće od t_{pHL} odnosno t_{pLH} .

Za isto logičko kolo stvarna kašnjenja su promenljiva!

Proizvođač ova vremena specificira za TIP logičkog kola.

Znači invertori koje je neki proizvođač napravio imaju ova MAKSIMALNA kašnjenja. SVI.

Ali svaki pojedinačno može da ima realna kašnjenja koja su različita od primerka do primerka.



5

Kombinacione mreže su digitalni sistemi kod kojih izlazi zavise samo od **trenutnog** stanja ulaza.

Trenutnog - kada su završeni svi „prelazni“ procesi, pošto se promena ulaznih signala neće odmah reflektovati na stanje izlaza.

Kada će izlazi postati važeći zavisiće od kašnjenja kroz pojedina logička kola.

Ta kašnjenja ponekad mogu da izazovu i neželjene efekte, odnosno neželjena stanja na izlazu.

To je pojava **lažnih nula i jedinica na izlazu**, čemu ćemo posvetiti posebnu pažnju u sintezi kombinacionih mreža.



6

Sinteza i minimizacija kombinacionih mreža

Mada se najčešće kod sinteze kombinacionih mreža pod **minimizacijom** podrazumeva minimizacija

- broja upotrebljenih logičkih kola,
- broja ulaza u logička kola

isto tako se može govoriti i o minimizaciji

- broja upotrebljenih **komponenti**
- ukupnog kašnjenja signala od ulaza do izlaza
- ...



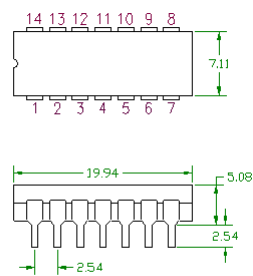
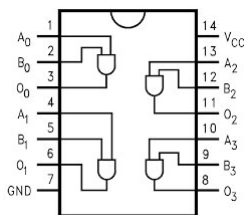
7

Standardna logička kola i delovi digitalnih sistema su raspoloživi u obliku integrisanih komponenti.

SSI - Small Scale Integration do 100 komponenti (tranzistora, dioda, otpornika, ...) odnosno 10 gejtova u jednom čipu

Gejt – standardno logičko kolo

Čip



inch=25.4mm
mils=inch/1000

14-Pin DIP

Sa stanovišta minimizacije treba uzeti u obzir ekonomski faktor. U većini slučajeva je ekonomski isplativije kupiti na primer 10000 integrisanih komponenti iste vrste, sa istim sadržajem, nego 5000+5000 komponenti integrisanih komponenti sa različitim sadržajem.



8

Način prikazivanja logičkih funkcija i direktna sinteza kombinacionih mreža

- Literal – logička promenljiva ili njena komplementna vrednost (A, \bar{A} , itd...)
- Proizvod – jedan literal ili logički proizvod dva ili više literala ($A, \bar{A}B, ABC$, itd...)
- Zbir - jedan literal ili logički zbir dva ili više literala ($A, \bar{A} + B, A + B + C$, itd...)
- Normalni proizvod – proizvod u kojem se promenljiva pojavljuje samo jednom bilo sa svojom pravom, bilo sa svojom komplementnom vrednošću
- Normalni zbir - zbir u kojem se promenljiva pojavljuje samo jednom bilo sa svojom pravom, bilo sa svojom komplementnom vrednošću
- Potpuni proizvod – za funkciju sa n promenljivih normalni proizvod u kojem se pojavljuju sve promenljive
- Potpuni zbir – za funkciju sa n promenljivih normalni zbir u kojem se pojavljuju sve promenljive
- Indeks potpunog proizvoda – vrednost n -bitnog binarnog broja dobijen tako što se u potpunom proizvodu, u kojem su promenljive uređenje na unapred definisan način, svaka promenljiva sa pravom vrednošću zameni binarnom jedinicom a svaka promenljiva sa komplementnom vrednošću binarnom nulom
- Indeks potpunog zbira – vrednost n -bitnog binarnog broja dobijenog tako što se u potpunom zbiru, u kojem su promenljive uređenje na unapred definisan način, svaka promenljiva sa pravom vrednošću zameni binarnom nulom a svaka promenljiva sa komplementnom vrednošću binarnom jedinicom
- Zbir proizvoda – logički zbir logičkih proizvoda
- Proizvod zbirova – logički proizvod logičkih zbirova



Na osnovu zahteva koji se postavljaju pred digitalni sistem uobičajeno je da se

1. Eventualno formira funkcionalna tabela digitalnog sistema
2. Eventualno napišu logičke funkcije
3. Prikaže digitalni sistem u obliku „električne“ šeme upotrebom simbola logičkih funkcija

Upotrebljen je pojam „eventualno“ pošto u sintezi mogu biti preskočeni neki koraci. Na primer bez pisanja logičkih funkcija korišćenjem Bulove algebre moguće je iz funkcionalne table prikazati digitalni sistem upotrebom simbola logičkih funkcija. Isto tako možda možemo da preskočimo i formiranje funkcionalne tabele, pa čak i pisanje logičkih funkcija i da direktno crtamo šemu.

Pojam električna šema je apsolutno ispravan pošto svakom simbolu logičkih funkcija odgovara realno logičko kolo sa svojim pravim električnim karakteristikama.

U ovom procesu sinteze se pojavljuje i među korak a to je „minimizacija“ ili prilagođenje logičkih funkcija zahtevima digitalnog sistema; na primer minimalan broj upotrebljenih tranzistora, minimalno kašnjenje, što manji broj kola iste vrste itd...



Neka je na primer zahtevima za digitalni sistem definisano

1. Digitalni sistem ima tri ulazna digitalna signala i jedan izlazni digitalni signal
2. Izlazni digitalni signal je na nivou logičke jedinice samo ako su dva ulaza digitalnih signala na nivou logičke jedinice.
3. U svim ostalim slučajevima izlazni digitalni signal je na nivou logičke nule

Prvi korak je da ulaznim i izlaznim digitalnim signalima dodelimo logičke promenljive. Ulaznim digitalnim signalima ćemo dodeliti logičke promenljive C, B i A, dok ćemo izlaznom digitalnom signalu dodeliti logičku promenljivu F. Radićemo u pozitivnoj logici tako da kada je na primer nivo ulaznog signal jednak logičkoj jedinici odgovarajuća logička promenjiva ima vrednost logičke jedinice.

Na osnovu zahteva koji su postavljeni pred digitalni sistem možemo formirati funkcionalnu tabelu.

C	B	A	F
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	0



Po pravilu u funkcionalnu tabelu se unose sve moguće vrednosti koje mogu da imaju ulazni signali odnosno logičke promenljive. Ponekad je moguće funkcionalnu tabelu napisati i u „skraćenoj“ formi, ali ona opet suštinski pokriva sve moguće situacije.

Neka je isti digitalni sistem sa tri ulaza i izlaza, ali je zahtev drugačiji.

Izlazni signal je na nivou logičke jedinice ako je bar jedan ulaz na nivou logičke jedinice.

C	B	A	F
0	0	0	0
b	b	1	1
b	1	b	1
1	b	b	1

U ovoj skraćenoj formi upotrebljen je simbol **b – bilo šta**, u smislu da ta logička promenjiva može uzeti i vrednost logičke jedinice i vrednost logičke nule, i da neće uticati na izlaznu logičku promenljivu.

Ponekad ćemo tu situaciju označavati i sa **X** sa istim značenjem.



Na osnovu osobina logičkih funkcija sada je iz funkcionalne tabele moguće napisati logičke funkcije. Posmatrajući kada funkcija ima vrednost logičke jedinice iskazi rečima osobina su redom od vrha funkcionalne table:

1. Funkcija ima vrednost logičke jedinice ako promenjiva C ima vrednost logičke nule I promenjiva B vrednost logičke jedinice I promenjiva A vrednost logičke jedinice
2. ILI - funkcija ima vrednost logičke jedinice ako promenjiva C ima vrednost logičke jedinice I promenjiva B vrednost logičke nule I promenjiva A vrednost logičke jedinice
3. ILI - itd...

Uočite da su iskazi povezani I i ILI logičkim funkcijama i da kada je potrebno da promenjiva ima vrednost logičke nule njena komplementna vrednost ima vrednost logičke jedinice. Prema tome možemo iz funkcionalne tabele napisati izraz

$$F = \bar{C}BA + C\bar{B}A + CBA\bar{A}$$

C	B	A	F
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	0



$$F = \bar{C}BA + C\bar{B}A + CBA\bar{A}$$

Izraz je dat u formi **zbira potpunih proizvoda** i uobičajen naziv je **normalna disjunktivna forma** ili **kanonička disjunktivna forma**.

Za formiranje izlazne funkcije smo upotrebili ILI logičko kolo zbog osobine da će na izlazu dati logičku jedinicu ako je bilo koji ulaz na logičkoj jedinici, a za uslove smo upotrebili I logičko kolo zbog osobine da na izlazu daje logičku jedinicu samo ako su svi ulazi na logičkoj jedinici.



Isto tako smo funkciju mogli posmatrati i kada njena vrednost ima vrednost logičke nule. Za formiranje izlazne funkcije ćemo upotrebiti I logičko kolo zbog osobine da će na izlazu dati logičku nulu ako je bilo koji ulaz na logičkoj nuli, a za uslove ćemo upotrebiti ILI logičko kolo zbog osobine da na izlazu daje logičku nulu samo ako su svi ulazi na logičkoj nuli. Pa su izrazi rečima počevši od vrha funkcionalne tabele:

1. Funkcija ima vrednost logičke nule ako promenjiva C ima vrednost logičke nule I promenjiva B vrednost logičke nule I promenjiva A vrednost logičke nule. Ovde je iskaz namerno napisan na ovaj način preko I logičkog kola, da bi bilo lakše praćenje. Međutim treba uočiti da je na „izlazu“ ovog iskaza aktivna logička nula odnosno da je u pitanju NI funkcija. Dualnost logičkih kola kaže da je to identično ILI logičkom kolu sa aktivnim logičkim nulama ili napisano preko jednačina Bulove algebre ovaj iskaz je $\overline{C\bar{B}\bar{A}} = C + B + A$. (da se podsetimo kada promenjiva treba da ima vrednost logičke nule njena komplementna vrednost ima vrednost logičke jedinice). Znači gledajući ILI logičku funkciju potrebno je da bi ona imala vrednost logičke nule da svi ulazi budu na nivou logičke nule.
2. I - funkcija ima vrednost logičke nule ako promenjiva C ima vrednost logičke nule I promenjiva B vrednost logičke nule I promenjiva A vrednost logičke jedinice
3. I - itd...

C	B	A	F
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	0

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



Drugi način tretirajući funkciju \bar{F} na isti način kako kada smo posmatrili da funkcija F ima vrednosti jedan (funkcija \bar{F} će imati vrednost logičke jedinice tamo gde funkcija F ima vrednost logičke nule)

$$\bar{F} = \bar{C}\bar{B}\bar{A} + \bar{C}\bar{B}A + \bar{C}B\bar{A} + C\bar{B}\bar{A} + CBA$$

$$\overline{(\bar{F})} = \overline{(\bar{C}\bar{B}\bar{A} + \bar{C}\bar{B}A + \bar{C}B\bar{A} + C\bar{B}\bar{A} + CBA)} = \overline{\bar{C}\bar{B}\bar{A}} \cdot \overline{\bar{C}\bar{B}A} \cdot \overline{\bar{C}B\bar{A}} \cdot \overline{C\bar{B}\bar{A}} \cdot \overline{CBA}$$

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

C	B	A	F	\bar{F}
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1



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

U oba slučaja dobijamo isti oblik funkcije koji je **proizvod potpunih zbirova** i naziva se **normalnom konjunktivnom formom** ili **kanoničkom konjunktivnom formom**.

Gledajući oba načina, na koji smo došli do funkcije, lako je uočiti formalizam da se napiše ovakav oblik funkcije: Vrsta u kojoj funkcija ima vrednost logičke nule pojavljuje se kao zbir promenljivih formiranih tako da ako promenljiva ima vrednost logičke nule u zbiru se pojavljuje sa pravom vrednošću, a ako ima vrednost logičke jedinice u zbiru se pojavljuje sa komplementnom vrednošću. Ovako formirani zbrovi se vezuju proizvodima u formiranju izlazne funkcije.



Oba načina prikazivanja funkcija možemo sada zapisati i na jednostavniji način korišćenjem definisanih indeksa proizvoda odnosno zbirova. Jasno je sada da ti indeksi predstavljaju vrednosti binarnih brojeva, kada stanja ulaznih promenljivih tretiramo kao binarne cifre, u odgovarajućoj vrsti. Znači

$$F = \sum_{C,B,A} (3,5,6)$$

$$F = \prod_{C,B,A} (0,1,2,4,7)$$

„cifra C je najveće težine, pa cifra B pa cifra A“

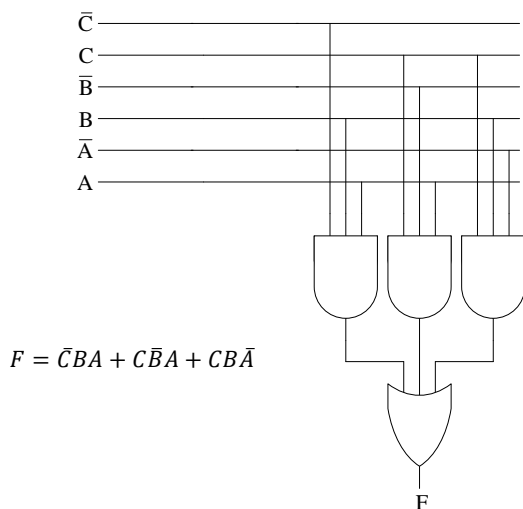
indeks	C	B	A	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Vrednost binarnog broja $b_{n-1}b_{n-2}b_{n-3}\dots b_1b_0$

$$B = b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + b_{n-3}2^{n-3} + \dots + b_12^1 + b_02^0$$

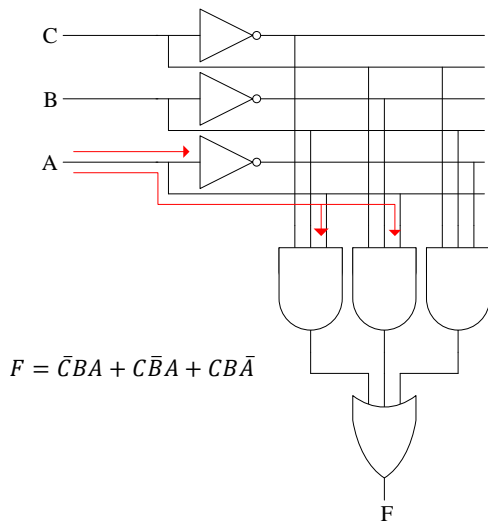


Električna, logička, šema funkcije F prikazane u obliku zbira proizvoda je



Da li su nam uvek na raspolaganju i prave i komplementne vrednosti signala. U većini praktičnih slučajeva NE.

Za formiranje električne, logičke, šeme će nam trebati i invertori pošto nam, na primer, trebaju komplementne vrednosti ulaznih signala jer raspoložemo samo sa pravim vrednostima.

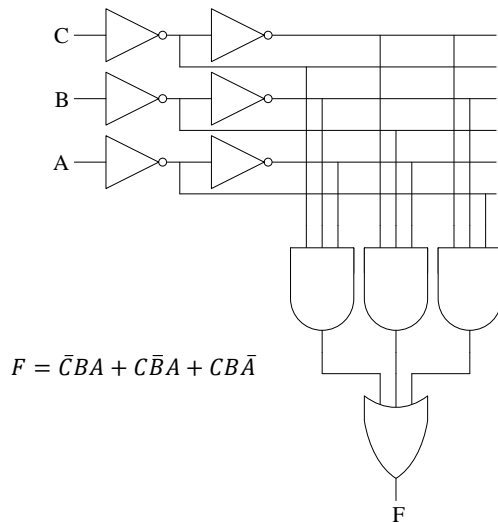


Primer: logičko kolo koje je napravilo promenljivu A „vidi“ tri ulaza u logička kola.

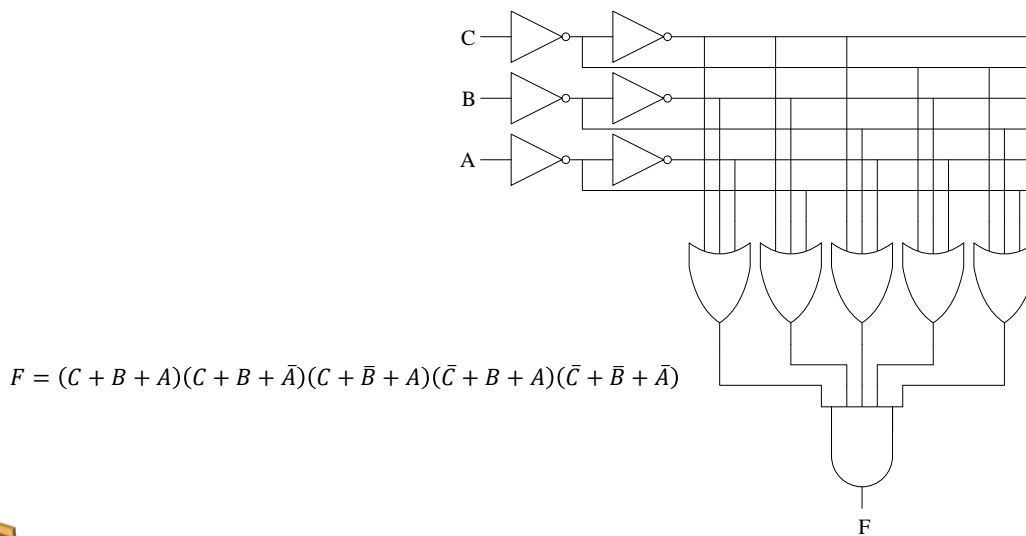
Sa stanovišta realnih logičkih kola to nije dobro.



Rasterećenje ulaza



Električna, logička, šema funkcije F prikazane u obliku zbira proizvoda je



Preporučuje se ovakav način crtanja i na ispitu.

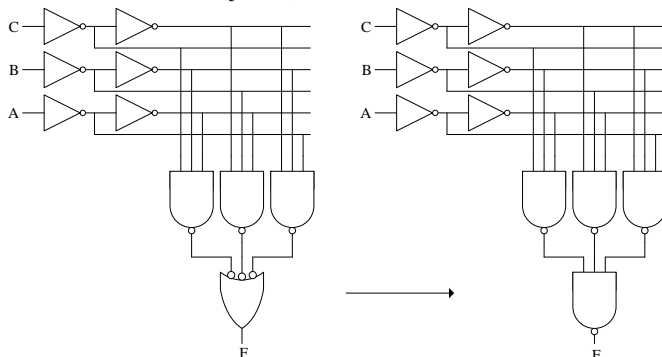
Kod crtanja električnih šema tačke kao spojevi se ne preporučuju. Ukrštanje dve linije u tom slučaju se ne smatra spojem.

Ono što treba uočiti

1. Na ovaj način je moguće realizovati bilo koju kombinacionu mrežu sa bilo kojim brojem ulaza i izlaza.
2. **Principski, funkcija realizovana kao zbir proizvoda i realizovana kao proizvod zbirova će zahtevati različiti broj resursa.** U ovom primeru funkcija realizovana kao zbir proizvoda zahteva 3 trouzlazna I kola i jedno trouzlažno ILI kolo odnosno zahteva $3 \times 3 + 1 \times 3 = 12$ ulaza. Funkcija realizovana kao proizvod zbirova zahteva 5 trouzlaznih ILI kola i jedno petoulazno I kolo, odnosno zahteva $5 \times 3 + 1 \times 5 = 20$ ulaza. Očigledno je u ovom primeru „isplativije“ realizovati funkciju u obliku zbira proizvoda. Principski koja će realizacija biti jednostavnija zavisi od broja logičkih jedinica i nula koje su pojavljuju u funkcionalnoj tabeli, koloni, za željenu funkciju F, kao i kako se pojavljuju.
3. Za realizaciju bilo koje kombinacione mreže su pored invertora dovoljna samo I i ILI logička kola.
4. Bilo koja kombinaciona mreža će, zanemarujući invertore, uvek biti realizovana u maksimalno dva nivoa, odnosno imati „minimalno“ kašnjenje.



Međutim zahvaljujući dualnosti logičkih kola, De Morganovim obrascima, moguće je doći i do još nekih rezultata. Ajde da vidimo prvo preko električnih šema. Da u postojećoj realizaciji „dodamo“ na izlaze logičkih kola prvog nivoa invertore odnosno kola prvog nivoa da u realizaciji pretvorimo u NI logička kola a u drugoj realizaciji u NILI logička kola. Da ne bi promenili funkciju na ulaze logičkih kola u drugom nivou ćemo takođe dodati invertore, odnosno napraviti ILI logičko kolo sa aktivnim logičkim nulama na ulazu u prvoj realizaciji (dualnost: ILI sa aktivnim logičkim nulama na ulazu jednako I sa aktivnom logičkom nulom na izlazu tj NI), dok ćemo dodavanjem invertora na ulaze I kola u drugoj realizaciji napraviti I logičko kola sa aktivnim logičkim nulama na ulazu (dualnost: I sa aktivnim logičkim nulama na ulazu jednako ILI sa aktivnom logičkom nulom na izlazu tj NILI)



Znači moguće je bilo koju kombinacionu mrežu realizovati, izuzimajući invertore samo pomoću NI logičkih kola. Ili generalnije ako inverter smatramo NI logičkim kolom sa jednim ulazom: Bilo koju kombinacionu mrežu možemo realizovati korišćenjem samo NI logičkih kola. Do istog rezultata smo mogli doći i preko izraza Bulove algebre i De Morganovih obrazaca:

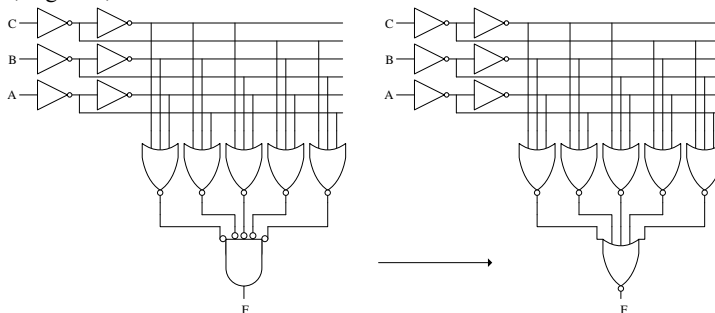
$$F = \bar{C}BA + C\bar{B}A + CB\bar{A} = \overline{\overline{\bar{C}BA + C\bar{B}A + CB\bar{A}}} = \overline{\bar{C}BA \cdot C\bar{B}A \cdot CB\bar{A}}$$



Za realizaciju funkcije u obliku proizvoda zbirova

$$\begin{aligned} F &= (C + B + A)(C + B + \bar{A})(C + \bar{B} + A)(\bar{C} + B + A)(\bar{C} + \bar{B} + \bar{A}) \\ &= \overline{\overline{(C + B + A)(C + B + \bar{A})(C + \bar{B} + A)(\bar{C} + B + A)(\bar{C} + \bar{B} + \bar{A})}} \\ &= \overline{\overline{(C + B + A)} + \overline{\overline{(C + B + \bar{A})}} + \overline{\overline{(C + \bar{B} + A)}} + \overline{\overline{(\bar{C} + B + A)}} + \overline{\overline{(\bar{C} + \bar{B} + \bar{A})}}} \end{aligned}$$

ili preko električnih, logičkih, šema



Znači moguće je bilo koju kombinacionu mrežu realizovati, izuzimajući invertore samo pomoću NILI logičkih kola. Ili generalnije ako inverter smatramo NILI logičkim kolom sa jednim ulazom: Bilo koju kombinacionu mrežu možemo realizovati korišćenjem samo NILI logičkih kola.



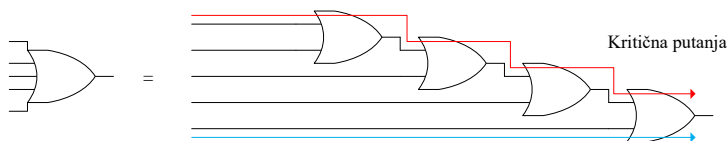
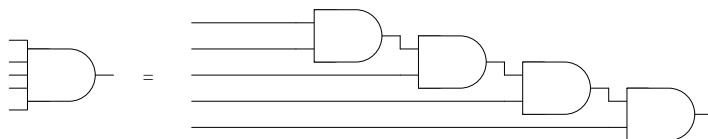
Ono što treba uočiti jeste da na ovaj način **nismo smanjili potreban broj logičkih kola**. Ukupan broj je ostao isti, osim što sada „kupujemo“ kola iste vrste, pa ćemo eventualno uštedi na broju čipova.

Međutim **ukupan broj ulaza je takođe ostao isti**, a na primer kod CMOS logičkih kola broj ulaza direktno diktira broj potrebnih tranzistora a time i površinu logičkih kola.

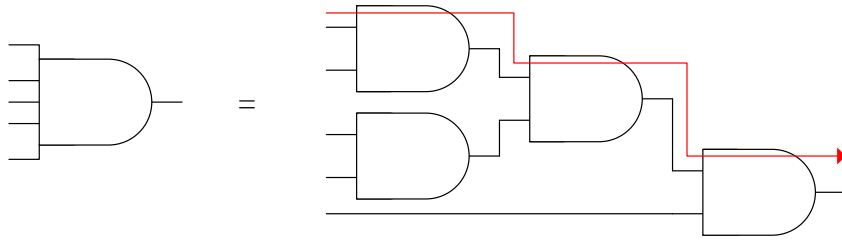
Unapred: radije bi radili sa NI logičkim kolima, pošto zauzimaju manju površinu.



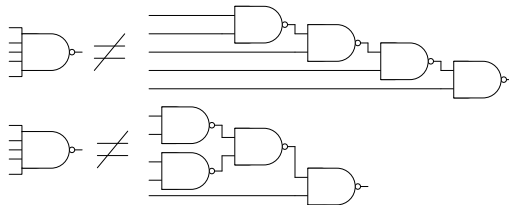
Postavlja se još jedno pitanje. Da li je na primer moguće realizovati bilo koju kombinacionu mrežu korišćenjem samo dvoulaznih logičkih kola. Odgovor je da moguće je, pri čemu treba biti oprezan u realizaciji višedulaznih logičkih kola korišćenjem dvoulaznih. Na slici su prikazani primeri i pravljenja invertora od dvoulaznih kola kao i način pravljenja višedulaznih kola u slučaju I i ILI logičkih kola.



Jasno je da sada kombinaciona mreža neće biti realizovana u „minimalnom“ broju nivoa, da će se broj nivoa povećati odnosno da će kritična putanja biti znatno duža, odnosno da će biti povećano kašnjenje. Ali kupujemo identična kola, odnosno gledano sa ekonomske strane kupujemo identične čipove. Možda je ekonomski isplativije, ako smemo da dozvolimo povećano kašnjenje. Međutim. Zašto treba biti oprezan? Istu realizaciju smo mogli da uradimo i sa manjim brojem logičkih kola i sa manjim kašnjenjem, znajući osobine logičkih funkcija.



Takođe, naročito treba biti oprezan sa kolima sa aktivnim logičkim nulama na izlazu NI i NILI.



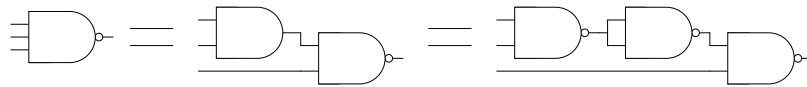
Isto važi i za NILI kola.



Na primeru trouzalnog NI kola preko jednačina i osobina Bulove algebre

$$F = \overline{CBA} = \overline{(CB)A}$$

što **sme**, i prikazano na šemi



Isto važi i za veći broj ulaza, kao i za NILI logička kola. Generalna zaključak je da je moguće bilo koju kombinacionu mrežu realizovati korišćenjem samo dvoulaznih NI logičkih kola, odnosno korišćenjem samo dvoulaznih NILI logičkih kola. Može i da se proširi, korišćenjem samo trouzlnih itd...



Minimizacija logičkih funkcija

U prethodnom delu nas je interesovala samo mogućnost realizacije kombinacionih mreža. Međutim, pojavom „digitalnih“ kola od interesa je postao i proces **minimizacije logičkih funkcija** koji bi trebao da obezbedi da se kombinaciona mreža realizuje ako je moguće sa što minimalnijim brojem komponenti. Međutim odmah da budemo jasni, pitanje je da li će to biti i ekonomski najisplativije, kao što smo već diskutovali. Kroz istoriju su realizovani mnogi algoritmi minimizacije, od kojih mnogi minimizuju logičke funkcije realizovane sa I i ILI logičkim funkcijama i minimizuju ukupan broj ulaza u logička kola. Sve ove tehnike minimizacije se zasnivaju na tome da se u funkciji datoj sa potpunim proizvodima ili zbiovima uoče proizvodi ili zbirovi čiji su **indeksi sa Hemingovim rastojanjem jedan** odnosno razlikuju se samo u jednom literalu. Primer

$$F = \dots + C\bar{B}\bar{A} + C\bar{B}A + \dots$$

Indeksi ova dva proizvoda su 4 i 6, odnosno 100 i 110. Hemingovo rastojanje (broj bita koji su različiti) je jedan, razlikuju se samo u drugom bitu. Razlikuju se samo po jednom literalu \bar{B} i B . U tom slučaju je moguće

$$F = \dots + C\bar{B}\bar{A} + C\bar{B}A + \dots = \dots + C\bar{A}(\bar{B} + B) + \dots = \dots + C\bar{A} + \dots$$

Znači umesto dva proizvoda za koja su nam trebala dva trouzlna I kola, dobili smo da nam treba samo jedno dvoulazno I kolo. Isto tako umesto da rezultat dva trouzlna I kola ide na izlazno ILI kolo, ići će rezultat samo jednog dvoulaznog I kola. Broj potrebnih kola smo smanjili sa dva na jedno, ali **ukupan broj ulaza smo smanjili sa $2 \times 3 + 2 = 8$ na $1 \times 2 + 1 = 3$** . U CMOS logici smo na primer **značajno smanjili broj potrebnih tranzistora odnosno površinu** koju bi nam zauzeo taj deo kombinacione mreže.



Mnogi algoritmi minimizacije su pogodni za mašinsku implementaciju ali nisu baš za ručnu sintezu.

Najpoznatiji algoritam za ručnu minimizaciju je minimizacija putem Karnoovih tabela, karti, koja je limitirana sa brojem ulaznih promenljivih za koje ima smisla raditi minimizaciju na taj način.

Do 4 ulazne promenljive jako jednostavno, radi se sve u 2D prostoru, sa 5 promenljivih se već prelazi u 3D prostor, sa 6 promenljivih u 4D prostor gde već postoji problem da se ispravno prati proces minimizacije, a za 7 već praktično postaje neupotrebljiv.



Osnova minimizacije putem Karnoovih tabela i jesu specifično napravljene tabele u kojima je lako uočiti proizvode ili zbrove sa Hemingovim rastojanjem 1.

Tabela sa 1 promenljivom

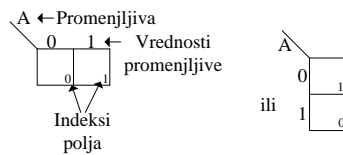
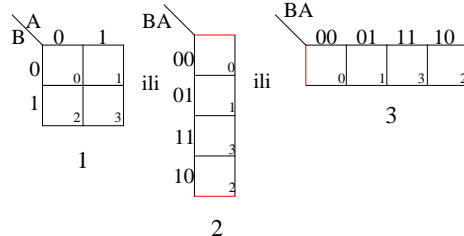


Tabela sa 2 promenljive



Uočite da indeksi kao i ranije odgovaraju vrednosti binarnog broja kojeg čine stanja promenljivih. Razlog zbog kojeg su indeksi odnosno stanja promenljivih prikazani na ovaj način je taj što sada susedna polja po indeksu imaju Hemingovo rastojanje 1. Susedna polja su ona koja imaju najmanje jednu zajedničku stranicu. (u 2D prostoru to je jedna, dok će u 3D i 4D prostoru biti više zajedničkih stranica) U 1. Primeru je to zadovoljeno bez razmeštaja stanja promenljivih, dok je u 2. i 3. praktično raspored promenljivih preuređen.



Tabela sa 3 promenljive

CB \ A	0	1
00	0	1
01		3
11	6	7
10	4	5

ili

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

Treba uočiti da su u 1. slučaju po definiciji susedna polja sa indeksima 0 i 4, odnosno 1 i 5, što vizuelno odgovara presavijanju tabele oko horizontalne ose po srediti i pravljenju prstena.

U 2. Slučaju su susedna i polja 0 i 2, odnosno 4 i 6 što se opet vizuelno dobija presavijanjem tabele oko vertikalne ose i pravljenjem prstena.



Tabela sa 4. promenljive

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

Isto kao i u prethodnim slučajevima, polja 0 i 2, 4 i 6, 12 i 14, 8 i 10 su takođe susedna. Vizuelno presavijanjem tabele oko vertikalne ose i ta polja će imati susedne stranice. Savijanjem oko horizontalne ose, a i po definiciji jediničnog Hemingovog rastojanja susedna polja su i 0 i 8, 1 i 9, 3 i 11, 2 i 10.



Tabela sa 5. promenljivih

DC \ BA		E=0				DC \ BA		E=1			
		00	01	11	10			00	01	11	10
00	0	1	3	2	00	16	17	19	18		
01	4	5	7	6	01	20	21	23	22		
11	12	13	15	14	11	28	29	31	30		
10	8	9	11	10	10	24	25	27	26		

Pored već definisanih susednih polja, susedna polja su i ona koja se u 3D prostoru dobijaju preklapanjem ove dve tabele, na primer 0 i 16, 1 i 17 itd...



Tabela sa 6. promenljivih

DC \ BA		FE=00				DC \ BA		FE=01			
		00	01	11	10			00	01	11	10
00	0	1	3	2	00	16	17	19	18		
01	4	5	7	6	01	20	21	23	22		
11	12	13	15	14	11	28	29	31	30		
10	8	9	11	10	10	24	25	27	26		

DC \ BA		FE=10				DC \ BA		FE=11			
		00	01	11	10			00	01	11	10
00	32	33	35	34	00	48	49	51	50		
01	36	37	39	38	01	52	53	55	54		
11	44	45	47	46	11	60	61	63	62		
10	40	41	43	42	10	56	57	59	58		

Pored već uvedenih susednih polja, susedna polja su i ona koja se u 4D prostoru dobijaju preklapanjem pojedinih tabela, na primer 0 i 32, 1 i 33, ..., 16 i 48, 17 i 49 itd...



Pre nego što pređemo na sam algoritam minimizacije uvedimo pojmove

Površina 0.-og reda - površina koja pokriva 1 polje u Karnoovoj tabeli.

Površina 1.-og reda - površina koja pokriva 2 susedna polja u Karnoovoj tabli.

Površina 2.-og reda - površina koja pokriva 4 susedna polja u Karnoovoj tabli.

Površina 3.-og reda - površina koja pokriva 8 susednih polja u Karnoovoj tabli.

Površina 4.-og reda - površina koja pokriva 16 susednih polja u Karnoovoj tabli.

Površina 5.-og reda - površina koja pokriva 32 susedna polja u Karnoovoj tabli.

Površina 6.-og reda - površina koja pokriva 64 susedna polja u Karnoovoj tabli.

Površine moraju biti pravilne u obliku pravougaonika, kvadrata, kvadra, kocke, hiperkvadra odnosno hiperkocke.



Uočite da smo na primer Tabelu sa 3 promenljive, mogli da zamislamo i u 3D prostoru kao preklapanje dve tabele

CB \ A	0	1
	00	0 1
01	2	3
11	6	7
10	4	5

=

B \ A	C=0	
	0	1
0	0	1
1	2	3

B \ A	C=1	
	1	0
0	4	5
1	6	7

Jasno je da su sada gledajući u 3D preklapanje, kao što smo već rekli susedna polja 0 i 4, 1 i 5 itd... Prema tome pravilna površina 1.og reda je i površina koja pokriva polja 0 i 4

CB \ A	0	1
	00	0 1
01	2	3
11	6	7
10	4	5

ili pravilna površina 2.og reda je i površina koja pokriva polja 0, 1, 4 i 5.

CB \ A	0	1
	00	0 1
01	2	3
11	6	7
10	4	5



Iz funkcionalne tabele, ili iz specifikacije kombinacione mreže, se u Karnoovu tabelu unose vrednosti funkcije za stanja ulaznih promenljivih u odgovarajuće polje.

Algoritam minimizacije

1. Da bi dobili funkciju u obliku zbira proizvoda potrebno je sve logičke jedinice koje se pojavljuju u Karnoovoj tabeli prekriti sa **što manjim brojem površina što višeg reda**.
2. Da bi dobili funkciju u obliku proizvoda zbirova potrebno je sve logičke nule koje se pojavljuju u Karnoovoj tabeli prekriti sa **što manjim brojem površina što višeg reda**.

Ako je ispunjen uslov da je broj površina minimalan sa najvećim mogućim površinama, algoritam garantuje da će i dobijene funkcije biti minimalne i da minimalnije od toga nije moguće realizovati funkcije korišćenjem I i ILI logičkih kola, podrazumevajući da raspolazemo i sa pravim i komplementnim vrednostima signala na ulazu u kombinacionu mrežu.



Ako se na nekom polju pojavi da funkcija može imati bilo koju vrednost oznaka b ili X to polje se može koristiti u minimizaciji ili kao logička jedinica ili kako logička nula kako bi se dobila što minimalnija forma. Na primer za sintezu u obliku zbira proizvoda može se koristiti kao logička jedinica, a za sintezu u obliku proizvoda zbirova kao logička nula ili obrnuto, ili ...

Površine predstavljaju objedinjavanje proizvoda kod kojih je moguće uraditi sažimanje na već opisan način. Prilikom formiranja funkcije u obliku zbira proizvoda u funkciji ostaju literali koji se ne menjaju na površini. Prilikom formiranja funkcije u obliku proizvoda zbirova u funkciji ostaju komplementi literala koji se ne menjaju na površini. Promenljive koje se menjaju na površinama se ne pojavljuju u proizvodima odnosno zbirovima.



Primer: Neka je na osnovu zahteva za kombinacionom mrežom napravljena Karnoova tabela

	BA			
DC	00	01	11	10
00	1	b	0	0
01	1	1	0	0
11	0	0	0	0
10	1	0	0	1

Da bi dobili minimalnu funkciju u obliku zbira proizvoda prekrivamo sve logičke jedinice površinama najvišeg reda, koristeći se i sa stanjem b

	BA			
DC	00	01	11	10
00	1	b	0	0
01	1	1	0	0
11	0	0	0	0
10	1	0	0	1

Pošto suštinski površine predstavljaju objedinjavanje proizvoda kod kojih je moguće uraditi sažimanje, u proizvodima ostaju literali koji se **ne menjaju** na površini.

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



Da bi dobili minimalnu funkciju u obliku proizvoda zbirova prekrivamo sve logičke nule površinama najvišeg reda, koristeći se i sa stanjem b

	BA			
DC	00	01	11	10
00	1	b	0	0
01	1	1	0	0
11	0	0	0	0
10	1	0	0	1

Pošto suštinski površine predstavljaju objedinjavanje proizvoda kod kojih je moguće uraditi sažimanje, u zbirovima ostaju komplementi literala koji se **ne menjaju** na površini.

$$F = (D + \overline{B})(\overline{D} + \overline{C})(\overline{D} + \overline{A})$$



Za realizaciju u obliku zbira proizvoda

$$F = \bar{D}\bar{B} + D\bar{C}\bar{A}$$

potrebno je 1 dvoulazno I kolo, 1 troulazno I kolo i 1 dvoulazno ILI kola, odnosno ukupan broj potrebnih ulaza je 7. Za realizaciju u obliku proizvoda zbirova

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

potrebno je 3 dvoulazna ILI kola i 1 troulazno I kolo, odnosno ukupan broj potrebnih ulaza je 9. Sa stanovišta klasične minimizacije je minimalnija realizacija u obliku zbira proizvoda.

U startu se ne zna koja je realizacija minimalnija. Treba izvesti obe pa na osnovu njihovih minimalnih izraza izvesti zaključak šta je zapravo minimalna realizacija.



Međutim sa stanovišta broja čipova, situacija može biti značajno drugačija ili ista ili ... Za realizaciju u obliku zbira proizvoda

$$F = \bar{D}\bar{B} + D\bar{C}\bar{A}$$

treba nam 4 invertora da bi napravili \bar{D} , \bar{C} , \bar{B} i \bar{A} što ima u 1 čipu (ima ih 6 u 14pinskom pakovanju, 14-2 za masu i napajanje = 12, dva po invertoru, 12/2=6) jedno dvoulazno I kolo što ima u 1 čipu, jedno troulazno I kolo takođe u 1 čipu i na kraju 1 čip sa dvoulaznim ILI kolima što je ukupno 4 čipa. Za realizaciju u obliku proizvoda zbirova

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

treba nam 4 invertora da bi napravili \bar{D} , \bar{C} , \bar{B} i \bar{A} što ima u 1 čipu, tri dvoulazna ILI kolo što ima u 1 čipu (ima ih 4 u 14pinskom pakovanju, 14-2 za masu i napajanje = 12, tri po dvoulaznom kolu, 12/3=4) i jedno troulazno I kolo takođe u 1 čipu što je ukupno 3 čipa. Sa ovog stanovišta je povoljnija realizacija u obliku proizvoda zbirova.



Isto tako algebarskim manipulacijama, na račun povećanog kašnjenja mogli smo realizaciju u obliku zbira proizvoda da uradimo i na sledeći način

$$F = \overline{D}\overline{B} + D\overline{C}\overline{A} = \overline{D}\overline{B} + (D\overline{C})\overline{A}$$

pa bi nam trebao jedan čip sa invertorima jedan čip sa dvoulaznim I kolima (treba nam ukupno tri i ima ih u jednom čipu) i jedan sa dvoulaznim ILI kolima što je ukupno 3.



Najčešća algebarska manipulacija da bi koristili kola iste vrste jeste da funkcije predstavimo tako da koristimo sa NI ili NILI logička kola. Za realizaciju samo sa NI logičkim kolima po pravilu je najlakše krenuti od zbira proizvoda

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

pa nam za realizaciju treba čip sa najmanje 4 invertora, 1 čip sa najmanje 2 dvoulazna NI kola i jedan čip sa najmanje jednim troulaznim NI kolom. Tri čipa ali različita. Za ovu realizaciju samo sa NI kolima ne postoji dokaz da je minimalna sa stanovišta klasične minimizacije. Možemo se samo nadati da je tako. Kako se u jednom čipu nalazi 4 dvoulazna NI kola, možemo ih iskoristiti kao invertore za potrebne 4 inverzije. I isto tako troulazno NI kolo na račun povećanog kašnjenja možemo realizovati kao $\overline{XYZ} = \overline{(XY)Z} = \overline{(\overline{XY})Z}$ pa bi nam za to trebalo 2 dvoulazana NI kola i 1 upotrebjeno kao invertor. Za ostatak nam treba još 2 dvoulazna NI kola. Ukupno je 9 dvoulaznih NI kola što možemo kupiti u 3 čipa, ali sada iste vrste, što je ekonomski isplativije, popust na količinu, od prethodne realizacije. Ali treba uočiti da smo kupili čip sa invertorima ostalo bi nam dva neupotrebljena. Za realizaciju funkcije sa dvoulaznim kolima nam treba 4 NI kola i jedan invertor. 4 NI kola imamo u jednom čipu a invertor možemo uzeti jedan od „viška“ koji nam je ostao. Znači možemo sve da realizujemo sa 2 čipa. Ostaje pitanje da li je jeftinije 1+1 ili 3. Ne postoji generalna odgovor.



Za realizaciju samo sa NILI logičkim kolima po pravilu je najlakše krenuti od proizvoda zbirova

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

pa nam za realizaciju treba pa nam za realizaciju treba čip sa najmanje 4 invertora, 1 čip sa najmanje 3 dvoulazna NILI kola i jedan čip sa najmanje jednim troulaznim NILI kolom. Tri čipa ali različita. I za ovu realizaciju samo sa NILI kolima ne postoji dokaz da je minimalna sa stanovišta klasične minimizacije. Možemo se samo nadati da je tako. Kako se u jednom čipu nalazi 4 dvoulazna NILI kola, možemo ih iskoristiti kao invertore za potrebne 4 inverzije. I isto tako jedno troulazno NILI kolo zameniti sa dva dvoulazna NILI kola i inverzijom $\overline{X + Y + Z} = \overline{(X + Y)} + \bar{Z} = \overline{(X + Y)} + \bar{Z}$, ukupan broji dvoulaznih NILI kola koja su nam potrebna je 10. Ukupan broj čipova 3. Ostaje na primer i da se proverii: pod uslovom da posedujemo invertovane vrednosti primenjivih i da ovu kombinacionu mrežu realizujemo kao jednostepeno složeno CMOS kolo koja realizacija bi bila minimalnija sa stanovišta površine koju zauzima. Probajte.

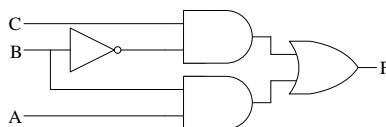


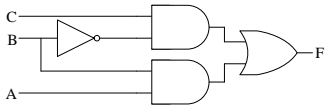
Pojava lažnih nula i jedinica

(race hazard)

Zbog kašnjenja koja su različita kod logičkih kola i zavisna i od fabrikacije i od temperature i od starosti komponente ... mogu se desiti neželjeni efekti na izlazu prilikom promene ulaznih signala. Najčešća situacija je da je izlaz na logičkoj jedinici, da se ulazni signali promene tako da izlaz treba da ostane na logičkoj jedinici ali se zbog propagacije signala pojavi kratkotrajna, lažna, nula na izlazu. Isto takva situacija i da je izlaz na logičkoj nuli, da se ulazni signali promene tako da izlaz treba da ostane na logičkoj nuli ali se zbog propagacije signala pojavi kratkotrajna, lažna, jedinica na izlazu. Dosta čest termin za ove kratkotrajne neželjene signale je glitch). Da vidimo na primeru. Neka je realizovana kombinaciona mreža

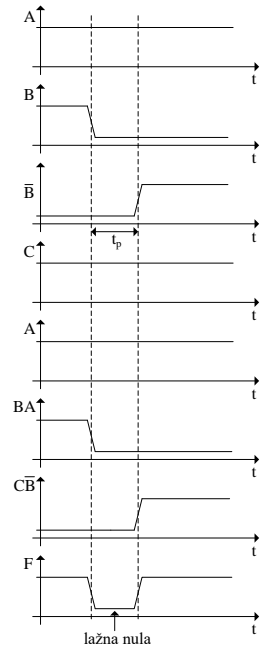
$$F = C\bar{B} + BA$$





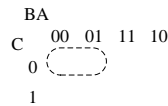
Da bi bolje uočili pojavu, ovaj put ćemo smatrati da su sva logička kola idealno brza, osim invertora koji ima kašnjenje t_p . Neka je stanje signala na ulazu $C = B = A = 1$. Za te ulazne signale $F = 1$. Posmatrajmo promenu samo signala B koji se u nekom trenutku promeni sa logičke jedinice na logičku nulu. Za stacionarno stanje signala $C = A = 1$ i $B = 0$ funkcija takođe ima vrednost $F = 1$. Znači bez obzira na promenu stanje izlaza treba da je logička jedinica.

ALI

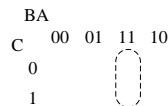


$$F = C\bar{B} + BA$$

U primeru je u pitanju funkcija sa tri promenljive C, B i A gde je član $C\bar{B}$ nastao minimizacijom površinom



a član BA površinom



odnosno funkcija je nastala minimizacijom potpunog oblika prikazanoj u Karnoovoj tabeli.

	BA			
C	00	01	11	10
0	1	1	1	0
1	0	0	1	0



	BA			
	00	01	11	10
C				
0	1	1	1	0
1	0	0	1	0

Ono što odmah možemo da uočimo jeste da smo posmatrali „prelaz funkcije“ sa jedne površine na drugu. Na prvoj uočenoj površini je $B = 0$ a na drugoj $B = 1$. Minimizacija je urađena prema pravilima ali površine nemaju zajedničkih polja ali imaju zajedničkih strana. U svim takvim slučajevima prelaz sa jedne na drugu površinu je po pravilu promena jedne promenljive (zbog ove zajedničke strane). I ta promena izaziva lažnu nulu, kao što smo videli na vremenskim dijagramima. Da bi se izbegla ova pojava mora se odstupiti od minimalne forme i ove dve površine „povezati“ dodatnom površinom kako bi se obezbedilo da imaju zajednička polja.

	BA			
	00	01	11	10
C				
0	1	1	1	0
1	0	0	1	0

U tom slučaju funkcija postaje $F = C\bar{B} + BA + CA$ i ovaj dodatni član koji povezuje površine obezbeđuje da se neće pojaviti lažna nula.



Potpuno identična situacija nastaje ako je funkcija prikazana u obliku proizvoda zbirova, odnosno minimizacija urađena prekrivanjem logičkih nula. Razlika je u tome što će se prilikom prelaska sa površine na površinu koje obezbeđuju logičke nule, pojaviti lažna jedinica. Rešenje je isto dodavanje zajedničke površine koja će povezati susedne površine koje nemaju zajedničkih polja.

Liči da smo na lak način razrešili problem, međutim u praksi je problem složeniji, pošto je moguće da se više ulaznih signala istovremeno menja. Osnovni problem koji nastaje jeste kada na kombinacionu mrežu gledajući samo I i ILI deo signali dolaze sa različitim kašnjenjem. Jedino moguće rešenje jeste da se ta kašnjenja ujednačavaju, dodavanjem dodatnih komponenti, invertora, ili da se preuređuje mreža i odstupa (kao što već jesmo) od „minimalne realizacije“ a što se u većini slučajeva radi heuristički ili simulacija. Suštinski presudno je razumevanje i iskustvo od strane projektanta.

