

Sinteza kombinacionih mreža

Kao što je u uvodnom delu rečeno, kombinacione mreže su digitalni sistemi kod kojih izlazi zavise samo od trenutnog stanja ulaza. Normalno, 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 ovih kombinacionih mreža. Mada se najčešće kod sinteze kombinacionih mreža govori o minimizaciji broja upotrebljenih kola, isto tako se može govoriti i o minimizaciji broja upotrebljenih komponenti ili o minimizaciji ukupnog kašnjenja kao što smo videli u nekim slučajevima. Standardna logička kola i delovi digitalnih sistema su raspoloživi u obliku integrisanih komponenti. 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.

Integrisana logička kola i digitalni sistemi

Najčešća podela integrisanih logičkih kola i digitalnih sistema je na osnovu količine tranzistora koji se nalaze u istom kućištu, čipu. Ta podela je na SSI, MSI, LSI, VLSI i ULSI integrisana kola.

SSI - Small Scale Integration do 100 komponenti odnosno 10 gejtova

MSI - Medium Scale Integratiom do 500 komponeti odnosno od 10 do 100 gejtova

LSI – Large Scale Integration do 300000 komponenti odnosno više od 100 do 10000 gejtova

VLSI – Very Large Scale Integration više od 300000 komponenti 10000 do 100000 gejtova

ULSI – Ultra Large Scale Integration više od 1500000 komponenti preko 100000 gejtova

Ova podela po vrednostima nije striktna i u literaturi se mogu naći i drugačiji brojevi, pri čemu današnja terminologija pod pojmom VLSI dizajna podrazumeva i VLSI i ULSI tehnologije.

Oznake koje su date u delu logičkih kola sa bipolarnim tranzistorima prvenstveno se odnose na kola u SSI tehnologiji i deo u MSI tehnologiji. Da ih ponovimo

Standardna oznaka je započinjala brojevima

1. 74 što je označavalo komercijalne komponente
2. 54 komponente predviđene za vojne primene

Sinteza kombinacionih mreža

Zatim su sledile slovne oznake koje su označavale familiju logičkih kola od koji su najznačajnije

1. Bez oznake – standardno TTL
2. L – Low power TTL
3. H – High power TTL
4. S – Schottky TTL
5. LS - Low power Schottky TTL
6. AS – Advanced Schottky TTL
7. ALS - Advanced low power Schottky TTL
8. F – Fast - oznaka koju je koristio proizvođač Fairchild a suštinski je ALS

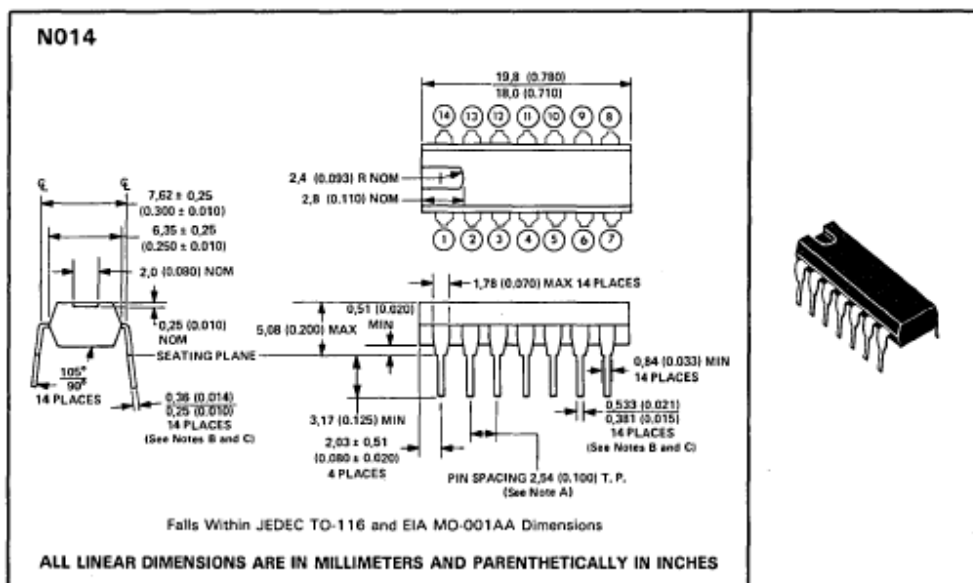
Ali isto tako su u istim kućištima, sa istim sadržajem pravljena i CMOS logička kola kod kojih su te oznake

9. C – CMOS verzija TTL kola – CMOS kompatibilni ulazi
10. HC – High speed CMOS – CMOS kompatibilni ulazi
11. HCT - High speed CMOS – TTL kompatibilni ulazi
12. AC – Advanced high speed CMOS - CMOS kompatibilni ulazi
13. ACT - Advanced high speed CMOS - TTL kompatibilni ulazi

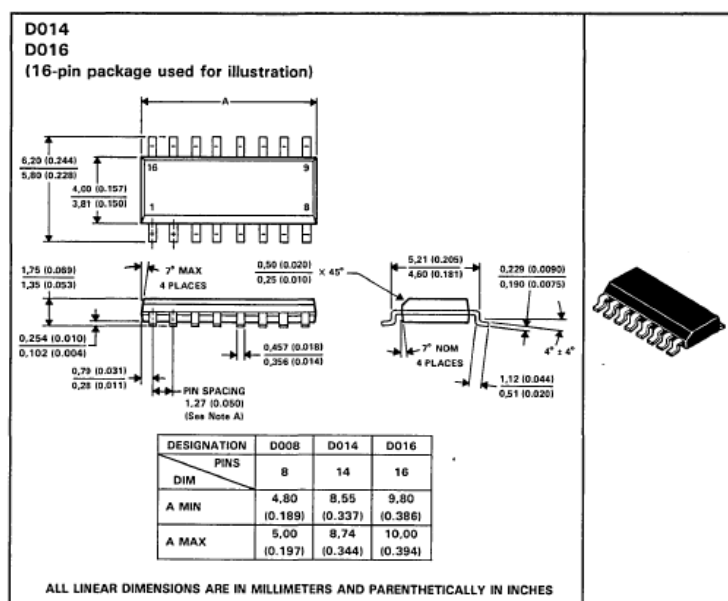
Mada su i jedne i druge komponente bile predviđene da rade sa standardnim naponom napajanja od +5V, nije bilo moguće njihovo direktno sprezanje zbog različitih nivoa logičkih jedinica i nula. Odnosno bilo je moguće CMOS na TTL, ali TTL na CMOS zbog nivoa logičke jedinice od +3.7V kod TTLa bi dovelo do toga da CMOS logička kola ne prepoznaju dobro logičku jedinicu. Zbog toga su i realizovane serije HCT i ACT gde su ulazi u CMOS logička kola prilagođena da mogu da prepoznaju nivoe TTL logičkih kola.

Primeri standardnih pakovanja u SSI tehnologiju su

DIP dual in package – DIL dual in line



SOIC – small outline integrated circuit



itd... Uočite da su prisutne i anglosaksonske mere koje su bile dominantne u istorijskom razvoju pakovanja integrisanih kola. Inč (inch) – 25.4mm, Mils – hiljaditi deo inča – 0.0254mm. Na primer standardna širina DIL pakovanja je 300milsa, itd...

Standardna pakovanja SSI tehnologije po pravilu imaju 14, 16 ili 20 nožica, pinova. Preko pinova se ostvaruje povezivanje putem veza na PCB (printed circuit board) sa ostalim komponentama. Uobičajeno je pin 7 (14 pinsko pakovanje), 8 (16 pinsko pakovanje) ili 10 (20 pinsko pakovanje) bio rezervisan za priključak mase, a pinovi 14, 16 i 20 za priključak napajanja. Ostali pinovi su rezervisani za ulaze i izlaze logičkih kola ili složenijih digitalnih sistema. Izbor tipa pakovanja zavisi od tehnologije u kojoj će PCB biti rađen. Proizvođači nude iste komponente u različitim pakovanjima i sve informacije o različitim dostupnim pakovanjima se nalaze u proizvođačkom opisu komponente.

Ono što je bitno jeste da su se u jednom pakovanju po pravilu nalazila kola iste vrste, na primer samo dvoulazna NI kola, ili samo invertori itd... Znači ako nam za sintezu kombinacione mreže treba dva dvoulazna NI kola, jedno troulazno NI, i jedan inverter morali smo kupiti tri čipa: sa dvoulaznim NI kolima, sa troulaznim NI kolima i sa inverterima. Ono što je takođe jako bitno jeste da se nekorišćeni ulazi i ulazi u nekorišćene delove komponente moraju postaviti na neaktivne nivoe. U suprotnom može doći do neželjenih efekata po rad aktivnog dela mreže. Kao što se vidi nije svejedno kako ćemo od početnih logičkih funkcija uraditi sintezu kombinacione mreže. U prethodnom primeru nadam se da ste uočili da sa dvoulaznim NI kolom možemo napraviti kratkim spajanjem ulaznih priključaka inverter – ne treba nam čip sa inverterima. Isto tako sa dva dvoulazna NI kola možemo napraviti jedno troulazno, normalno po cenu većeg kašnjenja $F = A \cdot B \cdot C = A \cdot (B \cdot C)$, Znači trebalo bi nam i manji broj čipova, a i kupovali bi čipove iste vrste. Opet, ako smemo da dozvolimo, kašnjenje će biti povećano pošto umesto jednog kašnjenja kroz troulazno logičko kola imamo kašnjenje kroz dva dvoulazna logička kola.

Način prikazivanja logičkih funkcija i direktna sinteza kombinacionih mreža

Radi lakšeg praćenja uvešćemo sledeće pojmove:

- Literar – 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 dobijenog 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

U uvodnom delu kursa smo rekli da 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 smo videli da svakom simbolu logičkih funkcija odgovara realno logičko kolo sa svojim pravim električnim karakteristikama.

Sinteza kombinacionih mreža

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 promenljiva 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. Primer pisanja funkcionalne tabele u skraćenoj formi

Neka je isti digitalni sistem sa tri ulaza i izlaza, ali je zahtev drugačiji. Izlani 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 promenljiva može uzeti i vrednost logičke jedinice i vrednost logičke nule, i da to 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

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

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 upotrebili I logičko kolo zbog osobine da će na izlazu dati logičku nulu ako je bilo koji ulaz na logičkoj nuli, a za uslove ćemo upotrebili 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 $\overline{\bar{C}\bar{B}\bar{A}} = C + B + A$. (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...

Gledajući ove iskaze, i ono što smo uočili u stavki 1, možemo principski doći na dva načina do izraza za funkciju F:

Prvi način

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

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

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

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

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

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

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 promenjiva 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

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

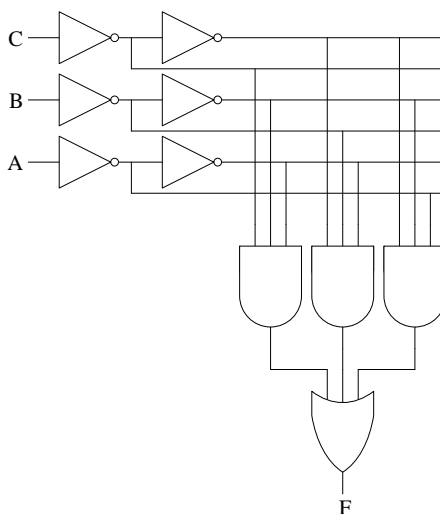
$$F = \prod_{C,B,A} (0,1,2,4,7)$$

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

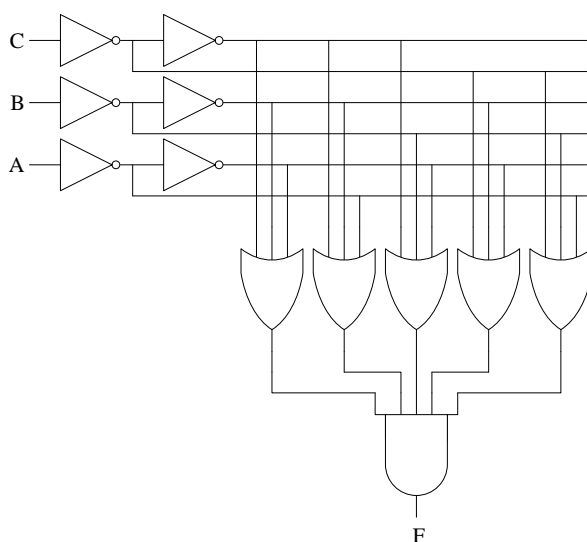
Za formiranje električne, logičke, šeme će nam trebati i invertori pošto nam trebaju komplemente vrednosti ulaznih signala. Da se podsetimo da smo u cilju rasterećenja prethodnih logičkih kola predvideli dva invertora povezana na red, tako da na njihovim izlazima dobijemo i komplementne i prave vrednosti signala, a da prethodna logička kola praktično više ulaz samo u jedan inverter. To smo nazvali rasterećenje ulaza, inače u suprotnom bi prethodna logička kola videla znatno veće kapacitivnosti pošto bi signal iz njihovog izlaza išao na ulaze znatno većeg i u principu „nepoznatog“ (zavisnog od funkcije) broja ulaza.

Sinteza kombinacionih mreža

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



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



Preporučuje se ovakav način crtanja i na ispitu. Ko crtanja električnih šema tačke kao spojevi se ne preporučuju. Ukrštanje dve linije u tom slučaju se ne smatra spojem.

Ono što treba uočiti

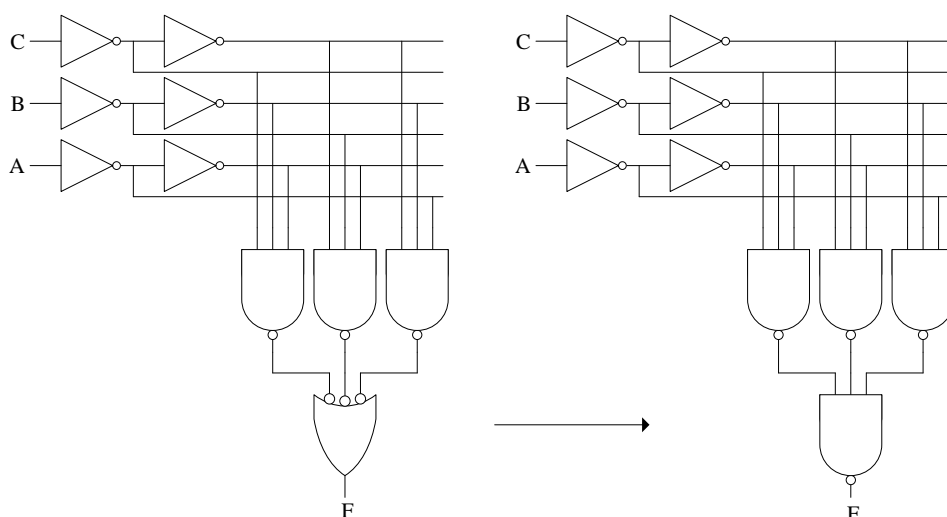
1. Na ovaj način je moguće realizovati bilo koju kombinacionu mrežu sa bilo kojim brojem ulaza i izlaza.
2. Principski funkcija realizovana kao zbir proizvoda i realizovana kao proizvod zbirova će zahtevati različit broj resursa. U ovom primeru funkcija realizovana kao zbir proizvoda zahteva 3 troulazna I kola i jedno troulazno ILI kolo odnosno zahteva $3 \times 3 + 1 \times 3 = 12$ ulaza. Funkcija realizovana kao proizvod zbirova zahteva 5 troulaznih 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

Sinteza kombinacionih mreža

koja će realizacija biti jednostavnija zavisi od broja logičkih jedinica i nula koje su pojavljuju u funkcionalnoj tabeli, koloni, za željenu funkciju F.

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:

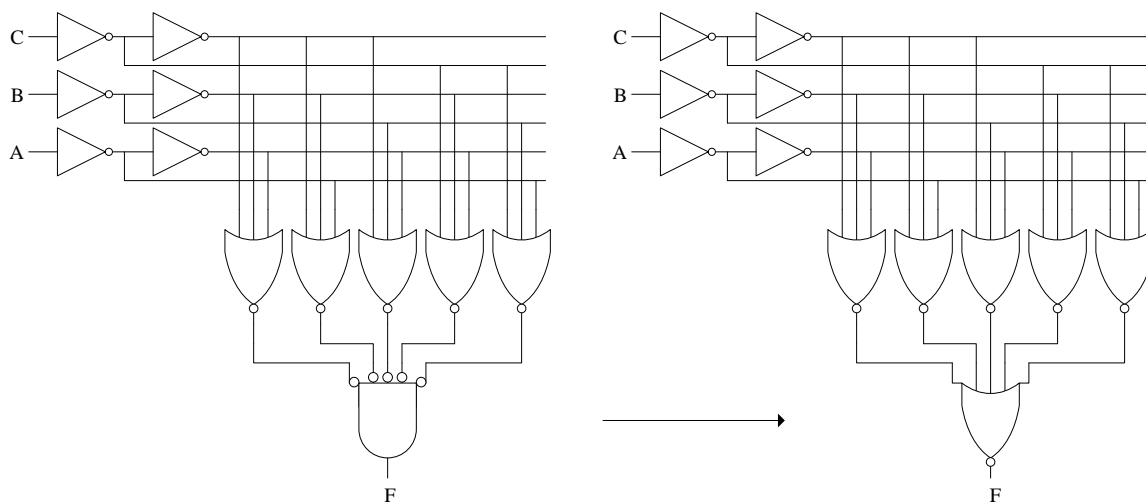
$$F = \bar{C}BA + C\bar{B}A + CB\bar{A} = \overline{\overline{\bar{C}BA} + \overline{C\bar{B}A} + \overline{CB\bar{A}}} = \overline{\bar{C}BA \cdot C\bar{B}A \cdot CB\bar{A}}$$

Za realizaciju funkcije u obliku proizvoda zbirova

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

Sinteza kombinacionih mreža

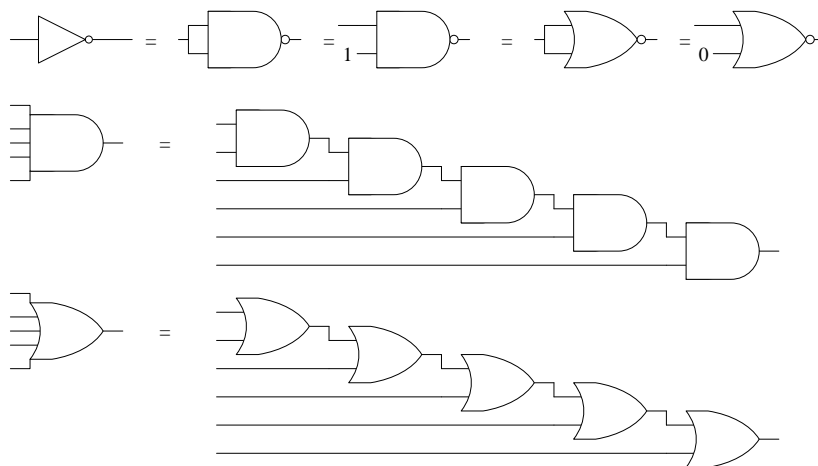
ili preko električnih, logičkih, šema



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.

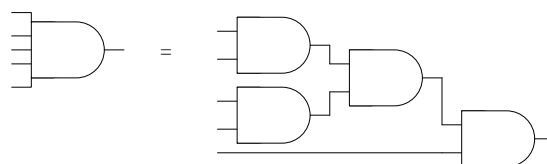
Ono što treba uočiti jeste da na ovaj način nismo smanjili potreban broj logičkih kola. Ukupan broj je ostao isti, osim što sada „kupujemo“ kola iste vrste, pa ćemo eventualno uštedi na broju čipova. Međutim ukupan broj ulaza je takođe ostao isti, a na primer kod CMOS logičkih kola broj ulaza direktno diktira broj potrebnih tranzistora a time i površinu logičkih kola. Da se podsetimo tada bi radije 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 višoulaznih logičkih kola korišćenjem dvoulaznih. Na slici su prikazani primeri i pravljenja invertora od dvoulaznih kola kao i način pravljenja višoulaznih kola u slučaju I i ILI logičkih kola.

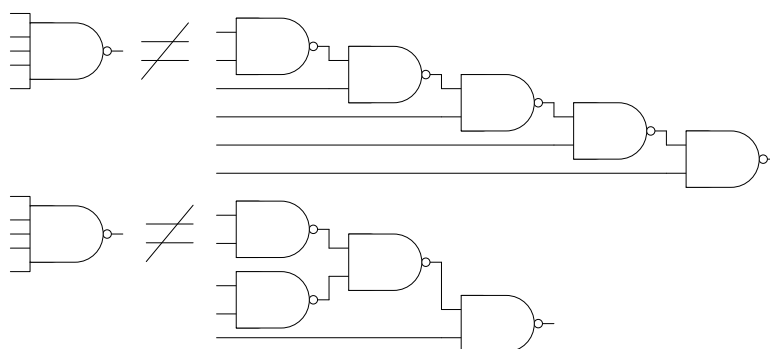


Sinteza kombinacionih mreža

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.



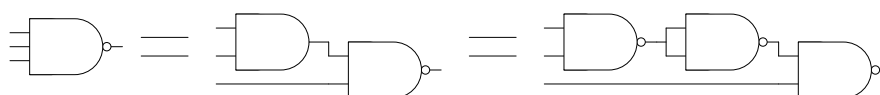
Takođe, naročito treba biti oprezan sa kolima sa aktivnim logičkim nulama na izlazu NI i NILI. Primer



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, ali 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 se 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 zbirovima uoče proizvodi ili zbrovi čiji su indeksi sa Hemingovim rastojanjem jedan odnosno razlikuju se samo u jednom literalu. Primer

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

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

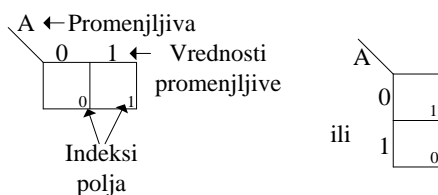
$$F = \dots + C\bar{B}\bar{A} + CB\bar{A} + \dots = \dots + C\bar{A}(\bar{B} + B) + \dots = \dots + C\bar{A} + \dots$$

Znači umesto dva proizvoda za koja su nam trebala dva dvoulazna I kola, dobili smo da nam treba samo jedno dvoulazno I kolo. Isto tako umesto da rezultat dva dvoulazna 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 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 (više sa teorijskog aspekta, praktično nema smisla)



Sinteza kombinacionih mreža

Tabela sa 2 promenljive (više sa teorijskog aspekta, praktično nema smisla)

		A	
		0	1
B	0	0	1
	1	2	3

ili

		BA	
		00	01
B	0	0	1
	1	2	3

ili

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

U 2. i 3. Primeru se već vidi osobina Karnoovih tabela. 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 drugom i trećem praktično raspored promenljivih jednak Grejovom kodu. Takođe treba uočiti u 2. i trećem primeru da su susedna polja i sa indeksom 0 i 2. Vizuelno kao kada bi presavili ovu 2D predstavu oko horizontalne ose u 2. primeru, oko vertikalne ose u 3. primeru i napravili prsten, pa bi onda i ta polja imala zajedničku stranicu.

Tabela sa 3 promenljive

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

ili

		BA			
		00	01	11	10
C	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 opet 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

		BA			
		00	01	11	10
DC	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.

Sinteza kombinacionih mreža

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

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

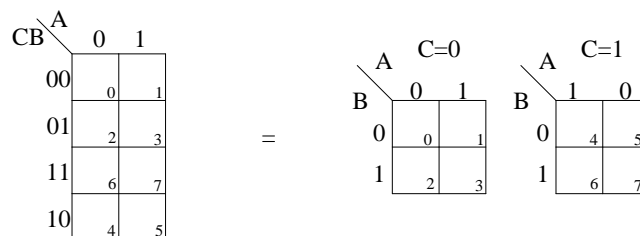
Površina 4.-og reda - površina koja pokriva 16 susednih polja u Karnoovoj tabli.

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

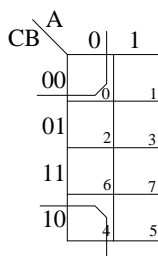
Sinteza kombinacionih mreža

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

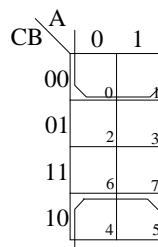
Površine moraju biti pravilne u obliku pravougaonika, kvadrata, kvadra, kocke, hiperkvadra odnosno hiperkocke. Uočite da smo na primer Tabelu sa 3 promenljive, mogli da zamislamo i u 3D prostoru kao preklapanje dve tabele



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



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



Identično važi i za tabele sa 4, 5 i 6 promenljivih.

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

Sinteza kombinacionih mreža

moguće realizovati funkcije korišćenjem I i ILI logičkih kola, podrazumevajući da raspoložemo 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

DC \ BA	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

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

Kao rezultat dobijamo

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

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

Sinteza kombinacionih mreža

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

Kao rezultat dobijamo

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

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

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. Kao što smo rekli 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

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

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

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

pa nam za realizaciju treba čip sa najmanje 4 invertora, 1 čip sa najmanje 2 dvoulazna NI kola i jedan čip sa najmanje jednim dvoulaznim 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 dvoulazno NI kolo na račun povećanog kašnjenja možemo realizovati kao $\overline{XYZ} = \overline{(XY)Z} = \overline{\overline{\overline{XY}}Z}$ pa bi nam za to trebalo 2 dvoulazna 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 neupotrebjena. 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 zbirava

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

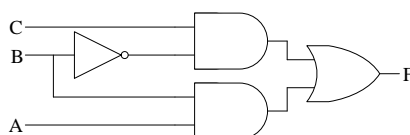
pa nam za realizaciju treba pa nam za realizaciju treba čip sa najmanje 4 invertora, 1 čip sa najmanje 3 dvoulazna NILI kola i jedan čip sa najmanje jednim dvoulaznim NILI kolom. Tri čipa ali različita. I za ovu realizaciju samo sa NILI kolima ne postoji dokaz da je minimalna sa stanovišta klasične minimizacije. 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 dvoulazno NILI kolo zameniti sa dva dvoulazna NILI kola i inverzijom $\overline{X + Y + Z} = \overline{(X + Y) + Z} = \overline{\overline{\overline{X + Y}} + Z}$, ukupan broji dvoulaznih NILI kola koja su nam potrebna je 10. Ukupan broj čipova 3. Ostaje na primer i da se proveriti: 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

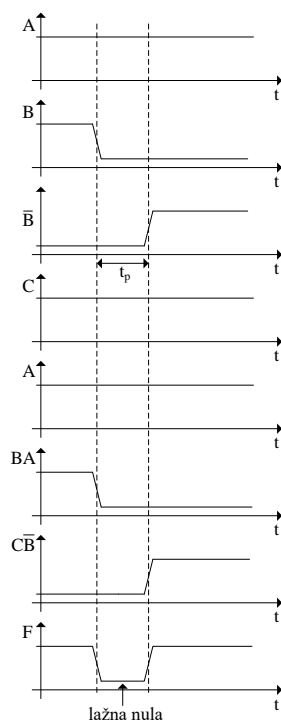
(race hazard)

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 glič (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. Međutim



Sinteza kombinacionih mreža

vidi se da će se na izlazu zbog kašnjenja kroz inverter pojaviti neželjeni signal odnosno, lažna nula. Ovakva situacija može da se u jednostavnim kolima uoči putem Karnoovih tabela. U primeru je u pitanju funkcija sa tri promenljive C, B i A gde je član $C\bar{B}$ nastao minimizacijom površinom

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

a član BA površinom

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

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

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

Do potpunog oblika funkcije smo mogli doći i algebarskim proširivanjem

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

Ali 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 zbrova, 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.

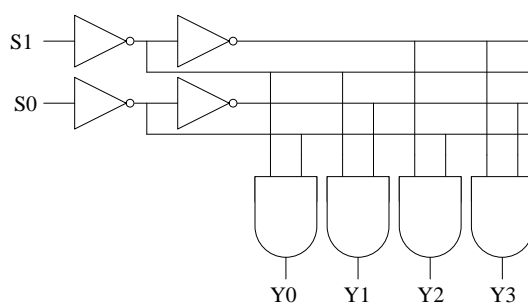
Liči da smo na lak način razrešili problem, međutim u praksi je problem složeniji, pošto je moguće da se više ulaznih signala istovremeno menja. Osnovni problem koji nastaje jeste kada na kombinacionu mrežu gledajući samo I i ILI deo signali dolaze sa različitim kašnjenjem. Jedino moguće rešenje jeste 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.

Kombinacione mreže srednjeg stepena integracije

Tokom istorijskog razvoja integrisanih kola i digitalne elektronike ustanovljeno je da neke konfiguracije kombinacionih mreža mogu lako da se iskoriste za realizaciju složenijih digitalnih sistema. Najčešće su dekoderi, koderi, multiplekseri, demultiplekseri, komparatori.

Dekoderi/demultiplekseri

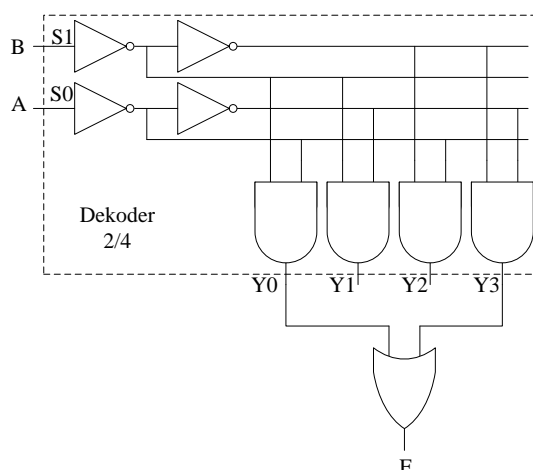
Pod pojmom dekodera podrazumevaju se kombinacione mreže sa n ulaza i m izlaza pri čemu je $n < m$. Najčešći su potpuni binarni dekoderi kod kojih je $m = 2^n$. Njihova struktura je u slučaju $n = 2$



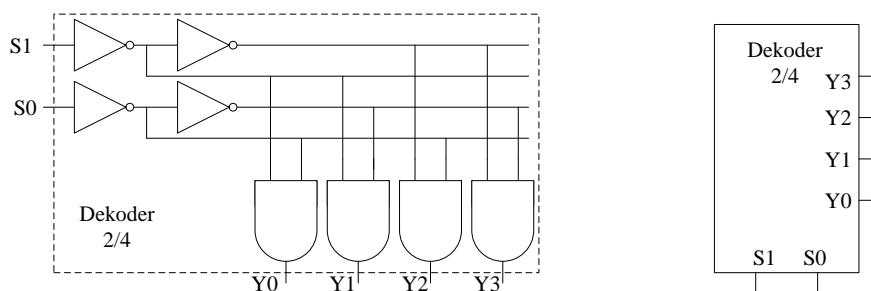
Ulazi S se nazivaju selekcionim ulazima a često i adresnim ulazima sa oznakom A. Ono što je njihova osnovna karakteristika je da na svojim izlazima daju potpune proizvode formirane od ulaznih signala. $Y_0 = \overline{S_1} \overline{S_0}$, $Y_1 = \overline{S_1} S_0$, $Y_2 = S_1 \overline{S_0}$ i $Y_3 = S_1 S_0$. Prikazani dekodere naziva se potpunim binarnim dekodrom 2 u 4 (2/4). Pošto raspolažemo potpunim proizvodima formiranim od ulaznih signala sada možemo lako realizovati funkcije korišćenjem samo dodatnih ILI logičkih kola.

Sinteza kombinacionih mreža

Na primer ako treba da realizujemo funkciju $F = \bar{B} \bar{A} + B A$

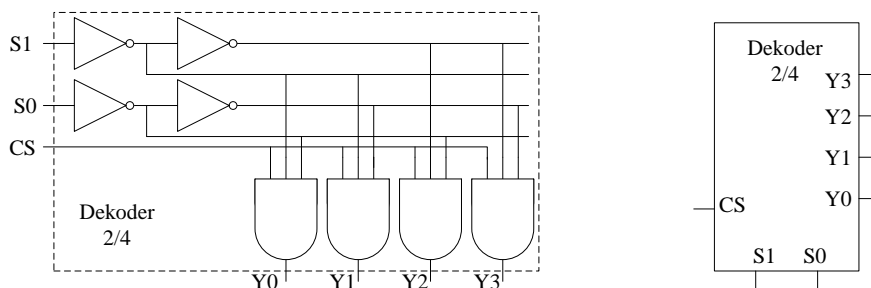


Simboli mreža srednjeg stepena integracije su pravougaoni sa svim označenim signalima. (Podvučeno je pošto na ispitu često zaboravite da označite signale unutar komponente, a onda nije moguće restaurirati šta ste i kako hteli da realizujete zadatak.)



Selekcioni signali su indeksirani po težini, dok su indeksi izlaza odgovarajući indeksima proizvoda koji se formira na tom izlazu. I to je standardno i podrazumeva se.

Da bi se na lak način omogućilo pravljenje mreža većih kapaciteta, sa većim brojem ulaza i izlaza uobičajeno komponente imaju dodatne selekcione signale koji „selektuju“ komponentu, uobičajeno nazvani CS (chip select).

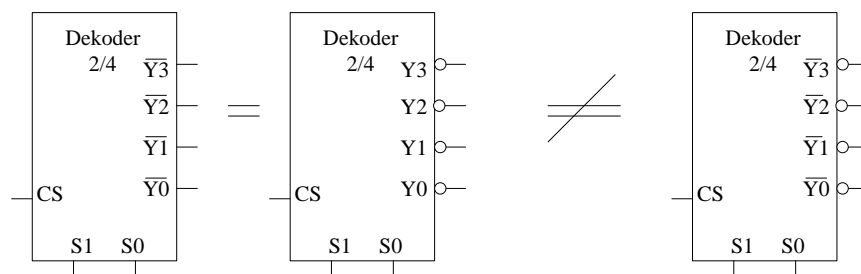


U ovom slučaju za dekodera se kaže da su mu aktivne logičke jedinice na izlazu, a CS signal dovodi sve izlaze u neaktivno stanje, odnosno kao što se vidi kada CS nije aktivan odnosno CS=0, svi izlazi će biti na neaktivnom nivou odnosno logičkoj nuli. Česta situacija je i da

Sinteza kombinacionih mreža

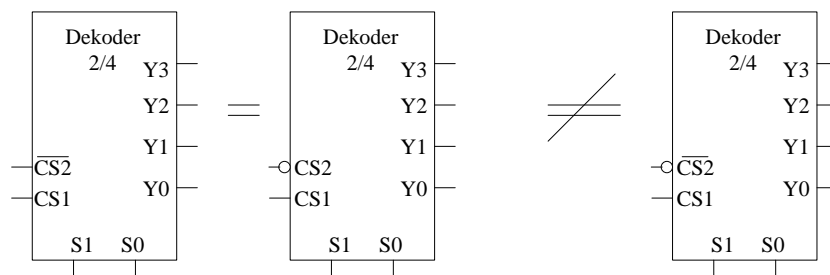
komponenta ima više od jednog CS signala. U tom slučaju su uvek povezani I logičkom funkcijom, odnosno svi moraju biti aktivni da bi se selektovala komponenta, $CS = CS1 \cdot CS2$.

Takođe ulazi i izlazi mogu biti i sa aktivnim nivoima logičke nule. Na primer da su izlazi sa aktivnim logičkim nulama



Obratite pažnju da su i prvi i drugi način označavanja isti, i mogu slobodno da se koriste, dok je treći pogrešan i takva komponenta nije isto što i prve dve. Suštinski treća komponenta bi opet bila dekodera sa aktivnim logičkim jedinicama na izlazu, postoji dvostruka negacija. Način prikazan na drugom simbolu se i najčešće koristi.

Primer sa više selekcionih signala.



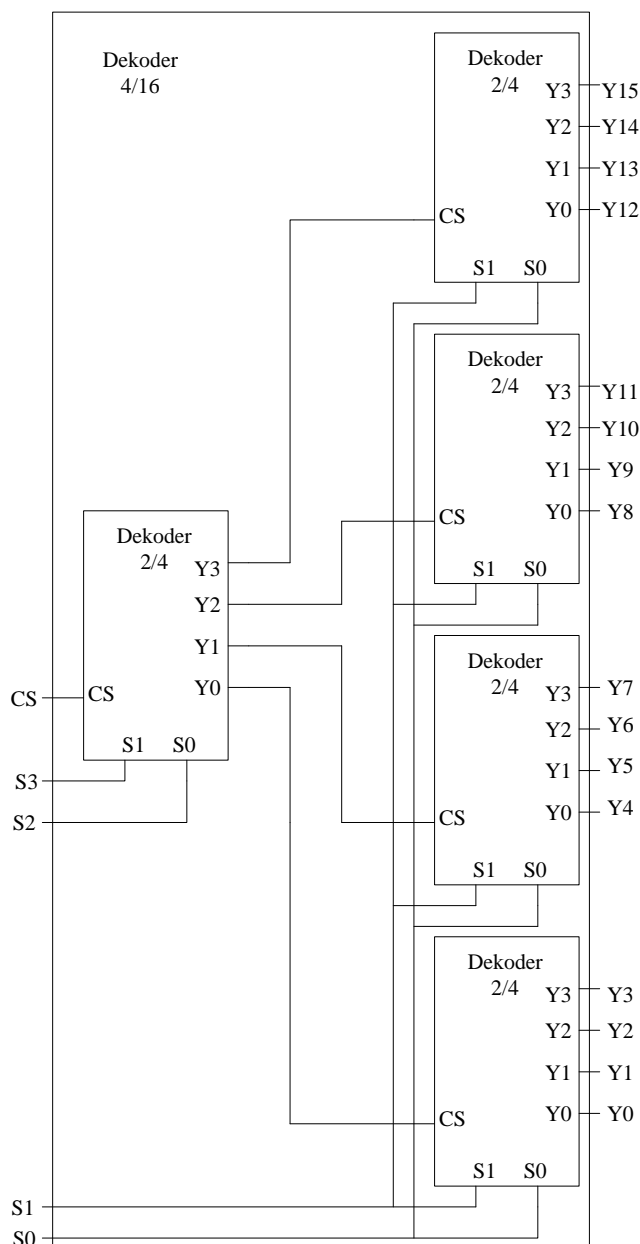
Unutrašnji selekcionni signal se formira kao $CS = CS1 \cdot \overline{CS2}$.

Izlazi komponente mogu biti sa otvorenim kolektorom, odnosno otvorenim drejnom. Ne postoji oznaka na simbolu, postoji informacija u opisu komponente. Takođe izlazi mogu biti sa stanjem visoke impedanse. Takođe ne postoji oznaka na simbolu, osim što se tada često umesto CS oznake, koristi E oznaka (enable). Ovo nije standardno i treba pročitati opis komponente.

Kao što je rečeno, selekcionni signali su korisni da bi se na lak način realizovale komponente većih kapaciteta. Na primer ako je potrebno realizovati dekodera 4/16 korišćenjem dekodera 2/4. Prvo što treba uočiti jeste da nam u „izlaznom“ nivou (tamo gde će biti krajnji izlazi nove komponente) u ovom slučaju treba $16/4=4$ dekodera. Da bi imali situaciju da je samo jedan izlaz aktivan moraju se selektovati pojedini dekoderi u izlaznom stepenu. To ćemo uraditi u prethodnom nivou. Samo selektovanje izlaznih dekodera je moguće uraditi nad dva načina: u piramidalnoj i u matričnoj strukturi.

Sinteza kombinacionih mreža

Piramidalna struktura. U ovom slučaju nam je dovoljan jedan selekcionni signal na komponentama

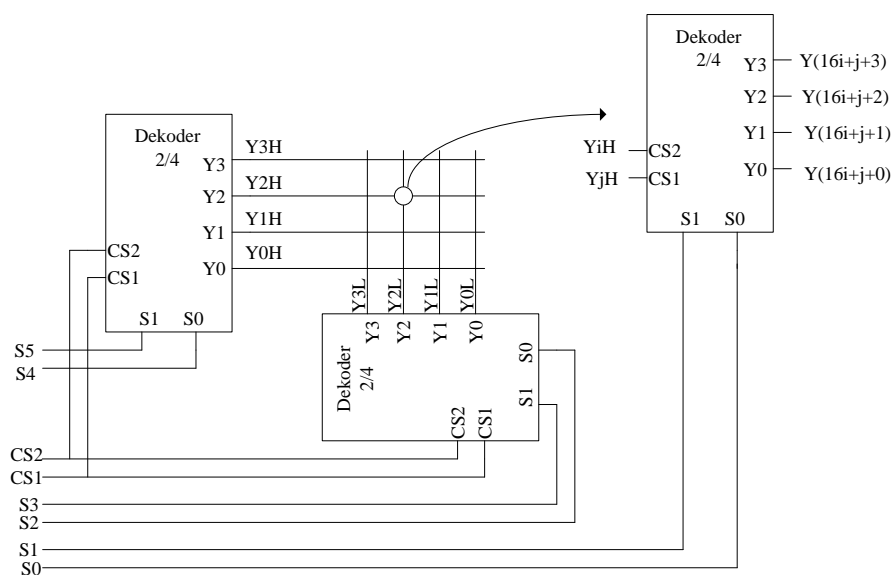


Mogući su i drugačiji načini povezivanja internih selekcionih signala. Prikazani način dozvoljava da se na lak način identifikuju indeksi novih izlaza. Po pravilu u izlaznom nivou se koriste selekcionni signali „najniže težine, a onda u prethodnim nivoima sve veće težine. Uočiti da na ovaj način možemo realizovati i komponentu sa većim brojem izlaza jedino će se povećavati broj nivoa. Na primer komponenta $6/2^6$ realizovana sa dekoderima $2/4$ bi imala tri nivoa. Naziv piramidalna struktura je zbog oblika koju ova struktura ima kao i načina dekodovanja.

Kod matrice strukture, treba nam dva selekcionna signala na komponentama u slučaju dvodimenzionog dekodovanja, odnosno tri u slučaju trodimenzionog dekodovanja itd... Za

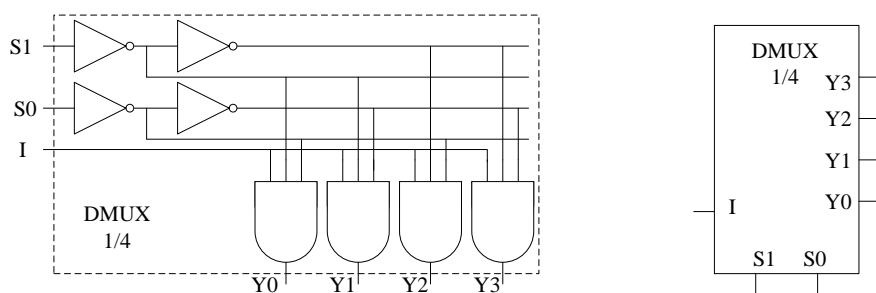
Sinteza kombinacionih mreža

veći broj izlaza matična struktura će imati prednost zbog manjeg broja upotrebljenih komponenti, ali kao što smo rekli potrebne su specifičnije komponente. U tom smislu da vidimo kako bi izgledala realizacija dekodera $6/2^6$ realizovana sa dekoderima $2/4$.



U izlaznom nivou se nalazi 16 dekodera $2/4$ (na slici je prikazan smo jedan) dok se jedan dekodere koristi za selekciju vrste (i) a drugi za selekciju kolone (j). Potreban broj komponenti nam je $16+1+1=18$. Način povezivanja internih selekcionih signala je identičan kao I kod piramidalne strukture. Uočiti da bi nam kod piramidalne strukture trebalo $16+4+1=21$ komponenta. Ova razlika postaje još značajnija kao su u pitanju komponente sa još većim brojem izlaza. Takođe uočiti da će matična struktura imati manje kašnjenje, pošto u ovom slučaju uvek su u pitanju dva nivoa, dok kod piramidalne strukture će broj nivoa zavistiti od veličine komponente. Ali za matičnu strukturu su potrebne komponente sa više selekcionih signala. Normalno moguće je praviti i kombinacije ove dve strukture.

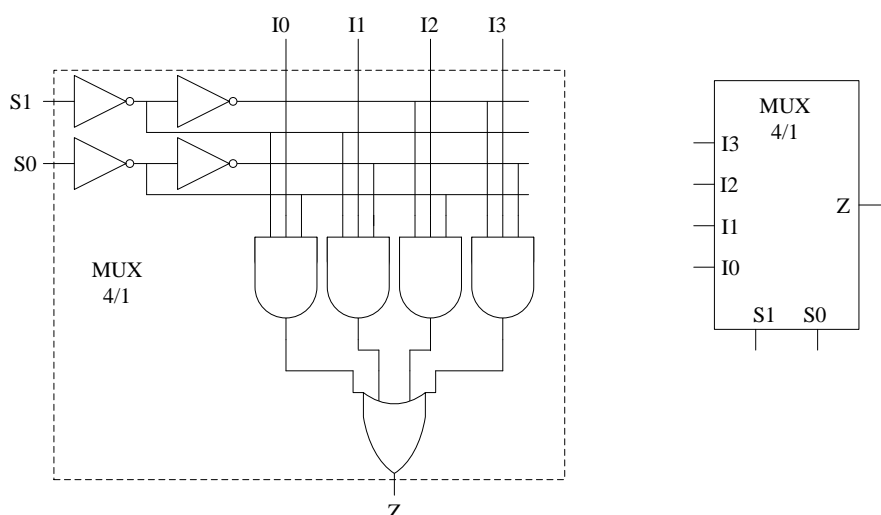
Demultipleksor je komponenta čija je osobina da sa ulaza (jednog) prosleđuje logički nivo na selektovani, adresirani, preko selekcionih signala, izlaz. Njegova unutrašnja struktura je



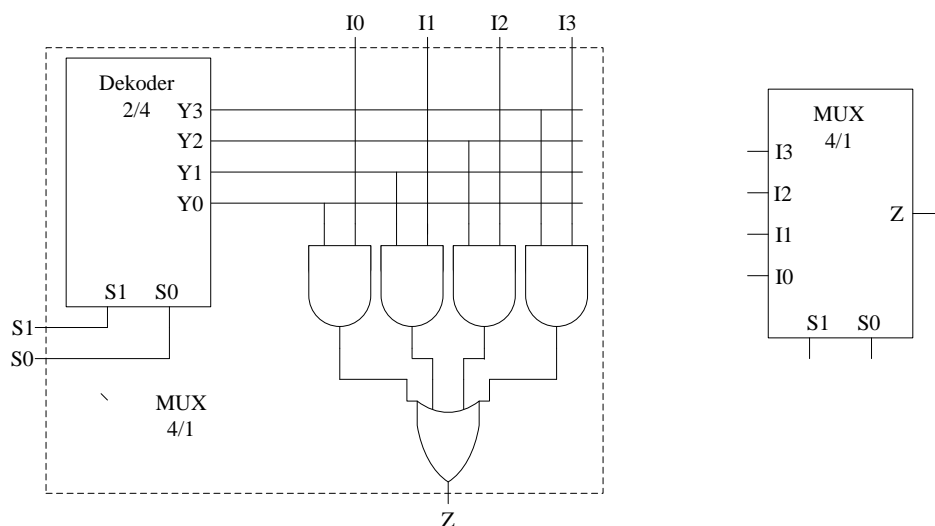
Kao što se vidi, njegova unutrašnja struktura je identična dekodera sa selekcionim signalom komponente. Zato se u katalogima dekodera i demultipleksori nalaze pod zajedničkim imenom dekodera/demultipleksor. Način pravljenja mreža većih kapaciteta je identičan kako i kod dekoderskih mreža.

Multiplekseri

Multiplekseri su specifične koderske mreže (broj ulaza veći od broja izlaza) koji imaju obrnutu funkciju od demultipleksera, odnosno ova komponenta prenosi stanje logičkog signala sa selektovanog, adresiranog, ulaza na izlaz. Znači ima n selekcionih signala, 2^n ulaza i jedan izlaz.

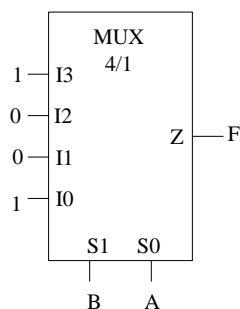


U nekim situacijama multiplekseri su komponente koje omogućuju još jednostavniju realizaciju kombinacionih mreža od dekodera. Na primer ako treba da realizujemo funkciju $F = \bar{B} \bar{A} + B A$. Pre nego što prikažemo realizaciju ono što treba uočiti da je multiplekser principski

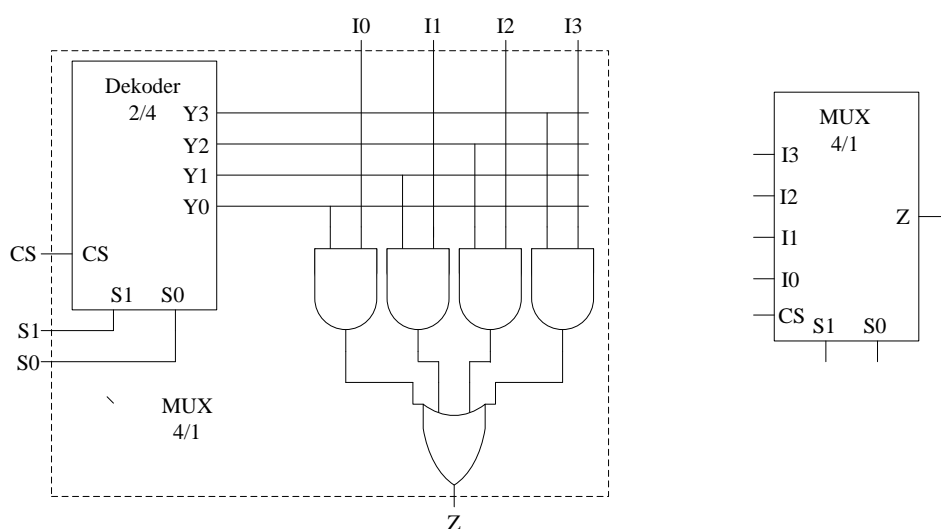


Sinteza kombinacionih mreža

odnosno da ponovo imamo selekciju potpunih proizvoda koje formiraju signali S1 i S0. U tom smislu jedino što treba da uradimo jeste da na ulaze multipleksera za odgovarajući proizvod dovedemo željene vrednosti funkcije



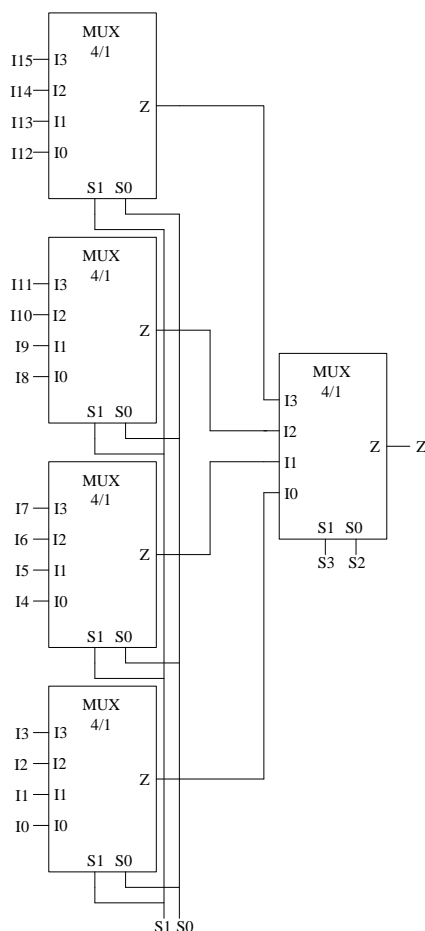
Kao i kod dekoderskih mreža komponenta može imati selekциони signal komponente



kako bi se dobile dodatne funkcionalnosti. Takođe i ulazi i izlazi mogu biti i sa aktivnim logičkim jedinicama (kao u prethodnim primerima) ali i sa aktivnim logičkim nulama. Izlaz može biti i sa otvorenim kolektorom, odnosno drejnom, ali takođe i sa stanjem visoke impedanse.

Multiplekseri su u VLSI kolima masovno koriste za pravljenje konfigurabilnih logičkih funkcija. Čest naziv tada im je „look up table“. Promenljive koje učestvuju u pravljenju funkcije se dovode na selekzione signale multipleksera, a rasporedom logičkih nula i jedinica na ulazu multipleksera se dobija željena funkcija. Ako je ova stanja na ulazima u multiplekser moguće menjati u toku rada ili definisati prilikom uključanja napajanja sistema, dobili smo lak način da konfiguriramo različite kombinacije mreže, logičke funkcije, u složenim digitalnim sistemima, bez potrebe da „projektujemo nov sistem“.

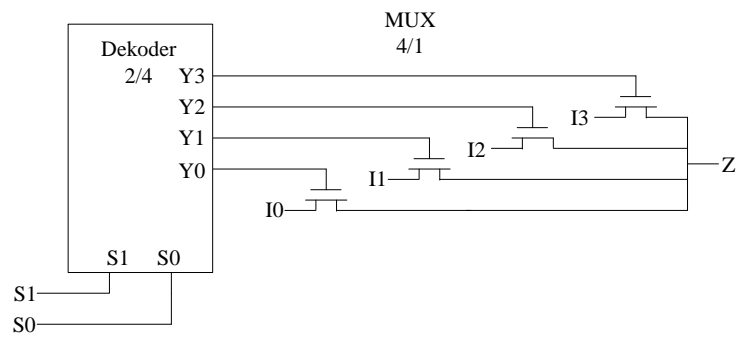
Primer kada sa multiplekserima 4/1 pravimo multiplekser 16/1. Ono što treba uočiti da prilikom pravljenja mreža većih kapaciteta, koje su po pravilu piramidalne strukture, selekcionni signal komponente nam nije neophodan.



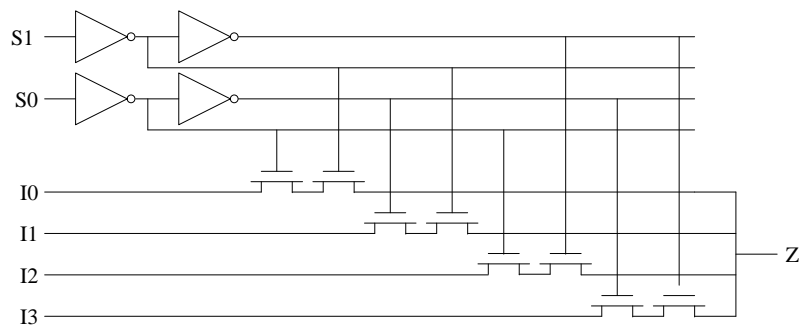
U razvoju digitalne elektronike dosta čest problem, lep za ispite, je bio da se na primer korišćenjem samo jednog multipleksera 4/1 i dodatnih logičkih kola realizuje funkcija sa 4 promenljive. U tom slučaju treba izabrati dve promenljive koje će se dovesti na selekzione signale multipleksera, dok će se sa preostale dve formirati funkcije koje se dovode na ulaze multipleksera. Pitanje je koje promenljive izabrati za selekzione a koje za pravljenje ulaza. Za odgovor može da pomogne minimalna forma funkcije u obliku zbira proizvoda. Gledajući tu minimalnu formu očigledno za selekzione promenljive treba izabrati one promenljive koje se u toj minimalnoj formi najčešće pojavljuju. U tom smislu sa preostale dve promenljive će nam biti jednostavnija realizacija logičkih funkcija za ulaze. Drugi način je da se gleda Karnoova tabela, pošto izbor selekcionih promenljivih u suštini predstavlja preklapanje polja u Karnoovoj tabeli površinama 2. reda. Suštinski, treba uočiti koje površine će imati najjednostavnije funkcije za kasniju realizaciju ulaznih logičkih funkcija. Međutim ovo se sa stanovišta sinteze kombinacionih mreža retko koristi. Setite se priče oko broja potrebnog broja čipova. Pogotovo što multiplekser pripada selekcionim mrežama koje se sa pass transistor logikom lako realizuju. Ili sa transmissionim gejtovima.

Sinteza kombinacionih mreža

Multiplekser u pass tranzistorskoj logici



a i sam dekoder u pass transistor logici i sve zajedno



Koder prioriteta

Osim multipleksera koji predstavlja specijalnu vrstu koderu u sintezi složenih digitalnih sistema (pogotovo u procesorskim) se najčešće koristi koder prioriteta. Osnovna njegova uloga jeste da koduje indeks ulaza na kojem se nalazi aktivan signal i najveće je vrednosti. Da bi bilo jasnije da vidimo funkcionalnu tabelu koderu prioriteta koji ima 4 ulaza. Da bi se kodovali indeksi potrebne su nam dve izlazne promenjive.

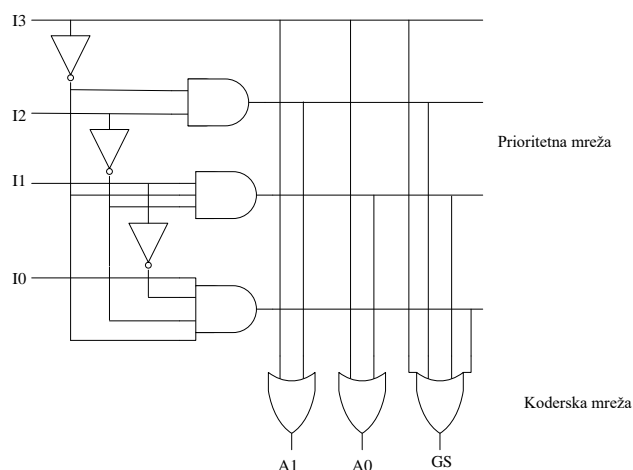
I3	I2	I1	I0	A1	A0
1	X	X	X	1	1
0	1	X	X	1	0
0	0	1	X	0	1
0	0	0	1	0	0
0	0	0	0	0	0

Neodređenost se pojavljuje u dve poslednje vrste. Preposlednja vrsta je situacija da je na ulazu sa najnižim indeksom aktivan signal, a poslednja da uopšte nema aktivnih signala. Iz tog razloga nam treba još jedan dodatni izlazni signal GS (group select) koji će pokazivati da li uopšte ima nešto da se koduje, odnosno da li postoji bilo koji aktivan ulazni signal

Proširena funkcionalna tabela u tom slučaju je

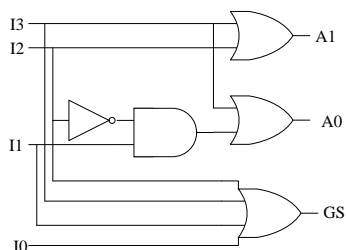
I3	I2	I1	I0	A1	A0	GS
1	X	X	X	1	1	1
0	1	X	X	1	0	1
0	0	1	X	0	1	1
0	0	0	1	0	0	1
0	0	0	0	0	0	0

Mada je moguće izvesti minimizaciju, u ovom slučaju je dosta jednostavnije direktno nacrtati realizaciju, ne trudeći se da bude baš minimalna. Direktno crtanje je moguće pošto su funkcije dosta „pravolinijske“.



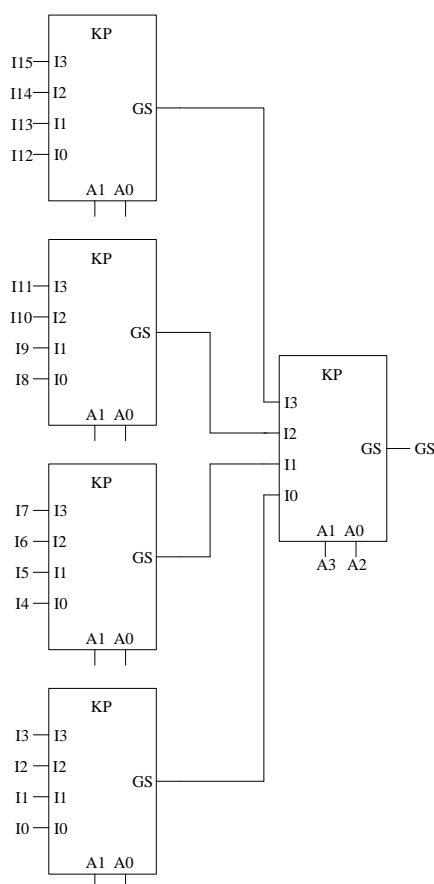
Sinteza kombinacionih mreža

Prioritetna mreža će obezbediti da se u unutrašnjosti komponente pojavi samo jedna aktivna logička jedinica sa onog ulaza koji ima aktivnu jedinicu i najvišeg je prioriteta odnosno ima najveći indeks. U tom slučaju je lako napraviti Koderski deo mreže. Može i minimalnije da se nacrti



međutim prva realizacija je lakša za proširivanje kada je potrebno realizovati veće mreže.

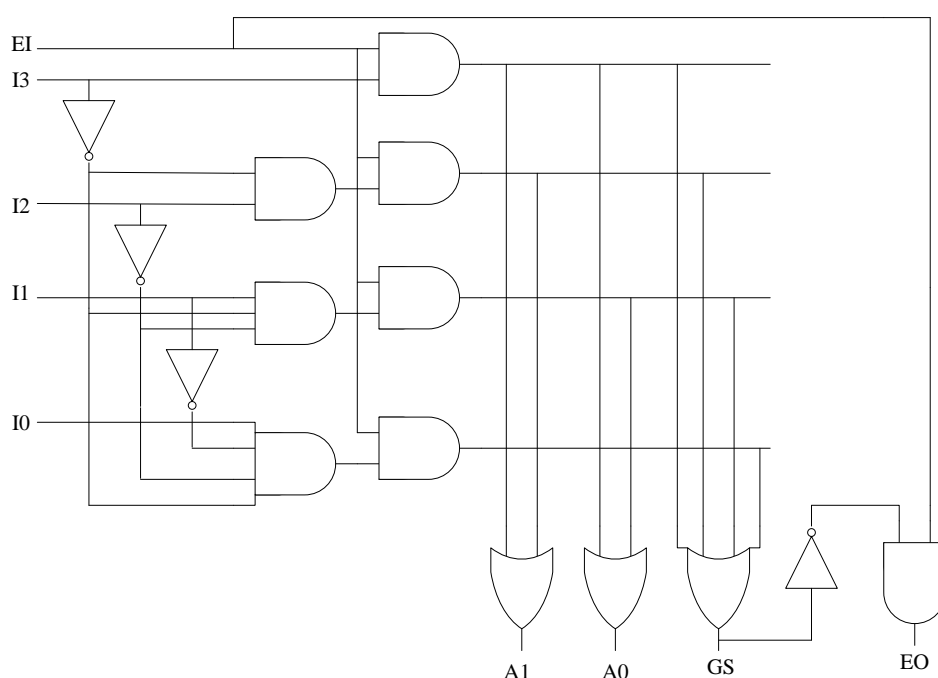
Prilikom pravljenja mreža većih kapaciteta, možemo se poslužiti idejama koje smo imali kod prethodnih kola. Na primer da je potrebno napraviti koder prioriteta sa 16 ulaza



Sa slike je vidljiva ideja da se iskoristi osobina koda prioriteta i u drugom nivou. Znači ako ima na bilo kojem koderu prioriteta u prvom nivou aktivnih logičkih jedinica, biće aktivan i njegov izlazni signal GS, pa će koder prioriteta u drugom nivou dati odgovarajući indeks koda prioriteta prvog nivoa (određen po prioritetu GS signala), što su u stvari signali A3 i A2 za „veliki“ koder. Ostaje problem kako odrediti signale A1 i A0. Ovde su u pitanju izlazi

pa njihovo direktno spajanje sigurno ne dolazi u obzir. A i logički njihovo direktno spajanje ne bi donelo validnu informaciju. U tom smislu bi morali na osnovne kodere prioriteta dodati još neke ulaze i izlaze, kako bi omogućili laku identifikaciju koji signali A1 i A0 su validni, a i da omogućimo njihovo lako logičko spajanje.

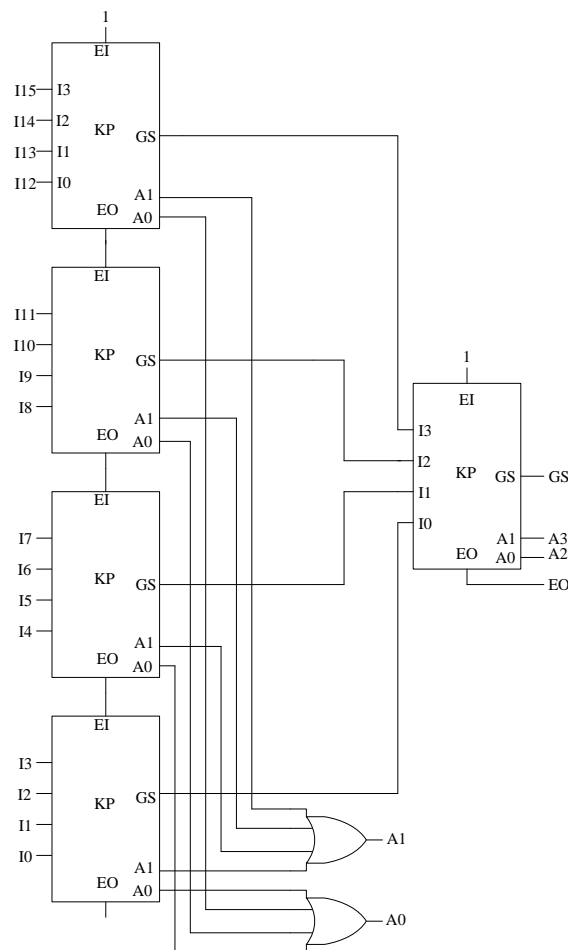
U tom smislu ćemo dodati ulazni signal EI (enable input) koji ako je aktivan, komponenta daje validne izlaze A1 i A0, a ako je neaktivan postavlja ih na logičke nule. Ovo možemo da uvedemo i za GS signal, u smislu da kada je on neaktivan komponenta nema šta da koduje bilo zato što nema aktivnih signala na ulazima, bilo zato što joj je „zabranjeno“ kodovanje. Kodovanje će joj biti zabranjeno ako na koderu višeg prioriteta ima šta da se koduje. Znači treba nam i jedan dodatan izlaz EO (enable output) koji suštinski dozvoljava koderu nižeg prioriteta da koduje. Dodavanjem ova dva signala u osnovnu šemu koderu prioriteta dobijamo



Možda ne prvi pogled liči da nam nije trebao EO signal i da smo umesto njega mogli upotrebiti invertovani GS signal, ne treba zaboraviti da informacija mora da se prenese „odgore na dole“ bez obzira šta se dešavalo u pojedinom koderu. Na primer sa ovako realizovanim GS on ima i značenje: nema šta da koduje pošto mu je zabranjeno, i da smo ga iskoristili kao EO on bi svojim invertovanim nivoom dozvolio koderu nižeg prioriteta da „radi“ a ne sme. Mogli smo da ne uslovima GS, to nam neće smetati za drugi novo koderu prioriteta, ali opet ne bi smeli da ga iskoristimo kao EO. To što on nema šta da koduje, ne znači i da oni „ispred njega“ po prioritetu nisu imali. Suština ovako realizovanog EO sa stanovišta pojedinog koderu je „dozvoljeno mi je da kodujem, nemam šta da kodujem, pa ću i ja dozvoliti koderu nižeg prioriteta, odnosno dozvoljeno mi je da kodujem imam šta da kodujem pa ću zabraniti nižim koderima da koduju, odnosno nije mi dozvoljeno da kodujem pa neću dozvoliti ni koderima nižeg prioriteta da koduju.

Sinteza kombinacionih mreža

Sa ovako realizovanim koderom prioriteta mreža sa 16 ulaza postaje

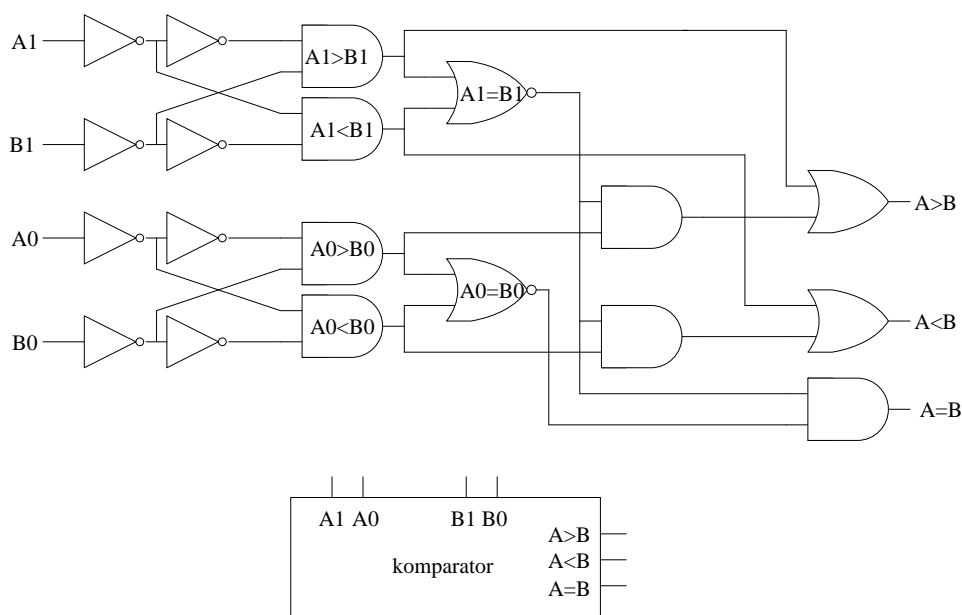


Mogli smo još da pojednostavimo mrežu u smislu da izlaze A1 i A0 kada je koder neaktivan dovedemo u stanje visoke impedanse. U tom smislu mogli bi na izlazima da izvršimo direktno spajanje, ne bi nam trebala četvoroulazna ILI kola.

Verovatno ste uočili da striktnih standarda za crtanje simbola kola srednjeg stepena integracije nema. Ono što je uobičajeno jeste da ulazi budu na jednoj strani simbola a izlazi na drugoj strani. Da se ne mešaju ulazi i izlazi na jednoj stranici simbola. I dosta često se ulazi crtaju sa leve strane simbola a izlazi sa desne strane simbola.

Komparatori

Videli smo da se prilikom izvođenja računskih operacija u binarnoj aritmetici često javlja potreba za poređenjem dva binarna broja. U tom smislu postoje standardne kombinacione mreže binarni komparatori koji kao ulaze primaju dva binarna broja A i B , a na izlazima daju rezultat poređenja $A < B$, $A = B$ i $A > B$. Isto kao i kod koda prioriteta moguće je funkcije pojedinih izlaza minimizirati, međutim i komercijalno raspoložive komponente a i ovde ćemo prikazati komponentu kod koje je lako pratiti putanje signala, kao i lako nadograđivati. Primer je poređenje dva neoznačena dvobitna binarna broja



Komponenta se lako može proširiti i realizovati sa 4 bita sledeći logiku koja je prikazana. Samu logiku možemo da posmatramo „odozgo na dole“ u smislu: Ako je $A_1 > B_1$ onda je $A > B$ i ne interesuju nas niži biti, odnosno ako je $A_1 < B_1$ onda je $A < B$ i ne interesuju nas niži biti, odnosno samo ako je $A_1 = B_1$ videćemo na nižim bitima šta se dešava. Ovo možemo da proširimo u opštem slučaju. Poredimo bite A_i, B_i samo ako su svi biti više težine bile jednaki, a u tom slučaju ako je $A_i > B_i$ onda je $A > B$ i ne interesuju nas niži biti, odnosno ako je $A_i < B_i$ onda je $A < B$ i ne interesuju nas niži biti, odnosno samo ako je $A_i = B_i$ videćemo na nižim bitima šta se dešava.

Ali ovu logiku možemo da posmatramo i „odole na gore“ Ako je $A_0 > B_0$ onda je možda $A > B$, odnosno ako je $A_0 < B_0$ onda je možda $A < B$, odnosno ako je $A_0 = B_0$ onda je možda $A = B$. Ovu informaciju prosleđujemo poređenju viših bita dok ne dođemo da bita najveće težine. Ovo možemo da proširimo u opštem slučaju. Poredimo bite A_i, B_i . Ako je $A_i > B_i$ onda je možda $A > B$ i ne interesuju nas šta je bilo na nižim bitima, odnosno ako je $A_i < B_i$ onda je možda $A < B$ i ne interesuju nas šta je bilo na nižim bitima. Pa tu

Sinteza kombinacionih mreža

informaciju prosleđujemo na više. Samo ako je $A_i = B_i$ prosleđujemo informaciju višim bitima u zavisnosti šta je bilo na nižim.

Možda je malo teže uočiti ovakav sled prenošenja informacija na istoj šemi, međutim ovo je značajno kada je potrebno dodati ulaze koji bi omogućili da se na lak način komponente iskoriste da bi se dobile mreže većih kapaciteta. Iz tog razloga se i formiraju nezavisno sva tri izlaza mada smo mogli na primer izlaz $A > B$ formirati kao nije $A < B$ i nije $A = B$ ili na sličan način $A = B$ kao nije $A < B$ i nije $A > B$ itd...

Upućujem Vas na komponentu 74HC85. Razmislite kako bi ti dodatni ulazi izgledali na nivou dvobitnog komparatora.