

Kola srednjeg stepena integracije



Katedra za elektroniku
prof dr Lazar Saranovac

Digitalna elektronika 1 - 2021/22

1

1

Integrirana logička kola i digitalni sistemi

Najčešća podela integriranih 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 integrirana kola.

SSI - Small Scale Integration do 100 komponenti odnosno 10 gejtova

MSI - Medium Scale Integratiom do 500 komponenti 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.



Katedra za elektroniku
prof dr Lazar Saranovac

Digitalna elektronika 1 - 2021/22

2

2

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.



3

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 nekorisćeni ulazi i ulazi u nekorisćene 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.

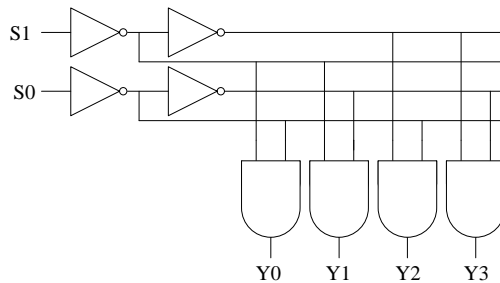
U prethodnom primerima nadam se da ste uočili da sa dvoulaznim NI kolom možemo napraviti kratkim spajanjem ulaznih priključaka inverter – ne treba nam čip sa inverterima. Isto tako sa dva dvoulazna NI kola možemo napraviti jedno troulazno, normalno po cenu većeg kašnjenja $F = A \cdot B \cdot C = A \cdot (B \cdot C)$, 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.



4

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 $n = 2$

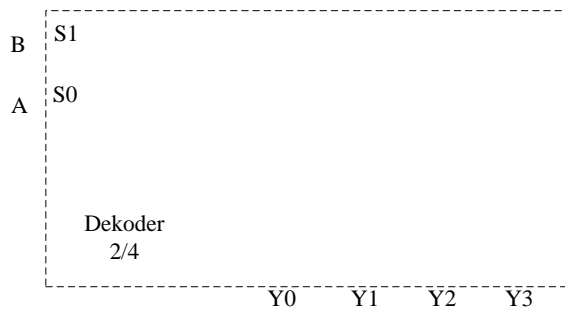


Ulazi S se nazivaju selekcionim ulazima a često i adresnim ulazima sa oznakom A . Ono što je njihova osnovna karakteristika je da na svojim izlazima daju potpune proizvode formirane od ulaznih signala. $Y_0 = \overline{S_1} \overline{S_0}$, $Y_1 = \overline{S_1} S_0$, $Y_2 = S_1 \overline{S_0}$ i $Y_3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



5

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



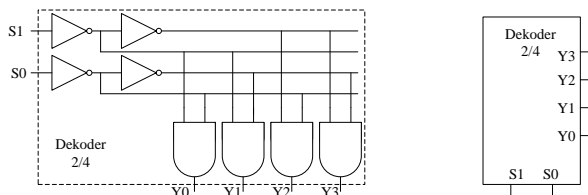
F



6

Simboli mreža srednjeg stepena integracije su pravougaoni sa svim označenim signalima.

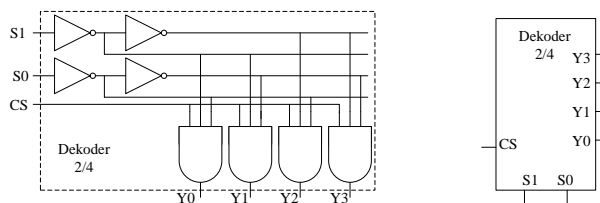
Podvučeno je pošto na ispitu često zaboravite da označite signale unutar komponente, a onda nije moguće restaurirati šta ste i kako hteli da realizujete zadatak.



Selekциони 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 selekzione signale koji „selektuju“ komponentu, uobičajeno nazvani CS (chip select).

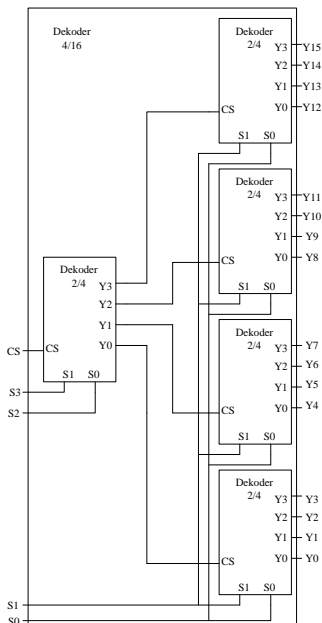


U ovom slučaju za dekodere se kaže da su mu aktivne logičke jedinice na izlazu, a CS signal dovodi sve izlaze u neaktivno stanje, odnosno kao što se vidi kada CS nije aktivan odnosno $CS=0$, svi izlazi će biti na neaktivnom nivou odnosno logičkoj nuli. Česta situacija je i da komponenta ima više od jednog CS signala. U tom slučaju su uvek povezani I logičkom funkcijom, odnosno svi moraju biti aktivni da bi se selektovala komponenta, $CS = CS1 \cdot CS2$.



Piramidalna struktura.

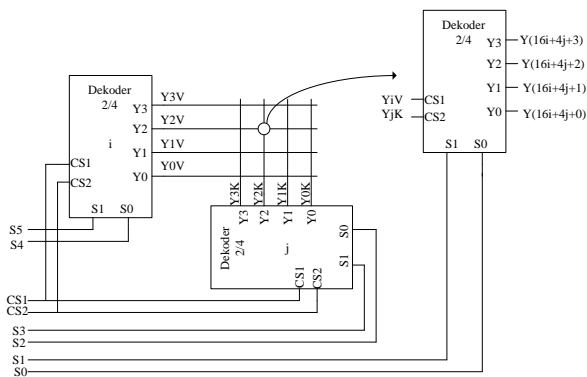
U ovom slučaju nam je dovoljan jedan selekcionni signal na komponentama



Mogući su i drugačiji načini povezivanja internih selekcionnih 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 6/2⁶ realizovana sa dekoderima 2/4 bi imala tri nivoa. Naziv piramidalna struktura je zbog oblika koju ova struktura ima kao i načina dekodovanja.



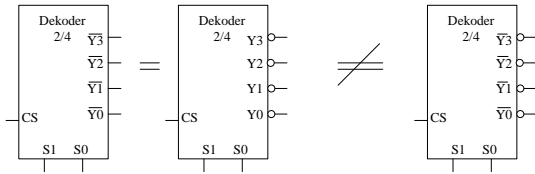
Kod matične strukture, treba nam dva selekcionna signala na komponentama u slučaju dvodimenzionog dekodovanja, odnosno tri u slučaju trodimenzionog dekodovanja itd... Za veći broj izlaza matična struktura će imati prednost zbog manjeg broja upotrebljenih komponenti, ali kao što smo rekli potrebne su specifičnije komponente. U tom smislu da vidimo kako bi izgledala realizacija dekodera 6/2⁶ realizovana sa dekoderima 2/4.



U izlaznom nivou se nalazi 16 dekodera 2/4 (na slici je prikazan smo jedan) dok se jedan dekodera koristi za selekciju vrste (i) a drugi za selekciju kolone (j). Potreban broj komponenti nam je $16+1+1=18$. Način povezivanja internih selekcionnih 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 zavisiti od veličine komponente. Ali za matičnu strukturu su potrebne komponente sa više selekcionnih signala. Normalno moguće je praviti i kombinacije ove dve strukture.

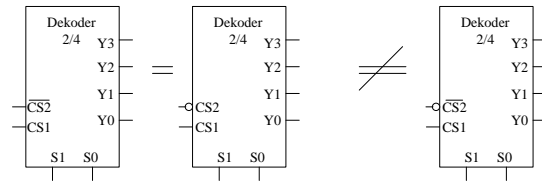


Takođe ulazi i izlazi mogu biti i sa aktivnim nivoima logičke nule. Na primer da su izlazi sa aktivnim logičkim nulama



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 dekoder sa aktivnim logičkim jedinicama na izlazu, postoji dvostruka negacija. Način prikazan na drugom simbolu se i najčešće koristi.

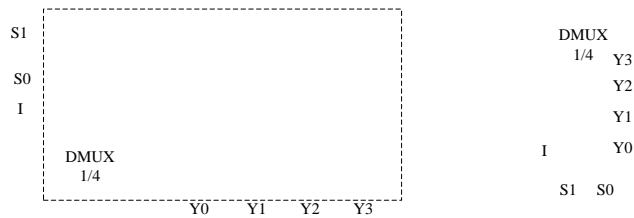
Primer sa više selekcionih signala.



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



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

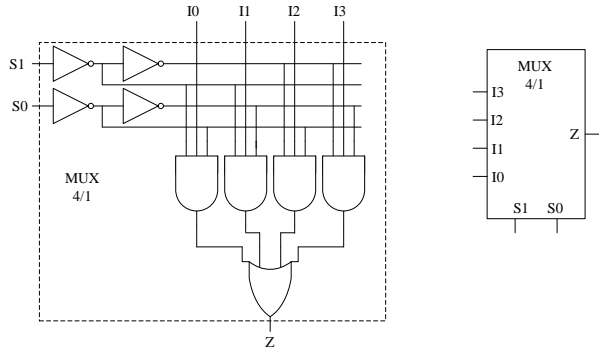


Kao što se vidi, njegova unutrašnja struktura je identična dekoderu sa selekcionim signalom komponente. Zato se u katalozima dekoderi i demultipleksori nalaze pod zajedničkim imenom dekoder/demultipleksor. Način pravljenja mreža većih kapaciteta je identičan kako i kod dekoderskih mreža.



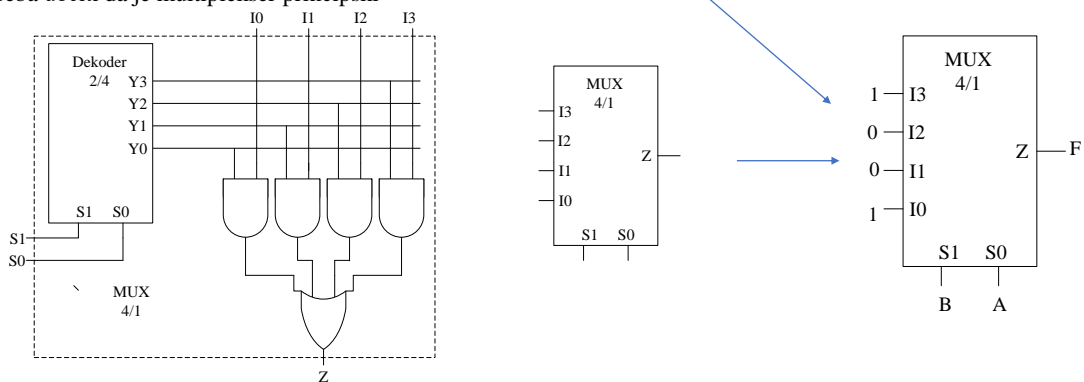
Multiplekseri

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.



13

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 prikazemo realizaciju ono što treba uočiti da je multiplekser principski

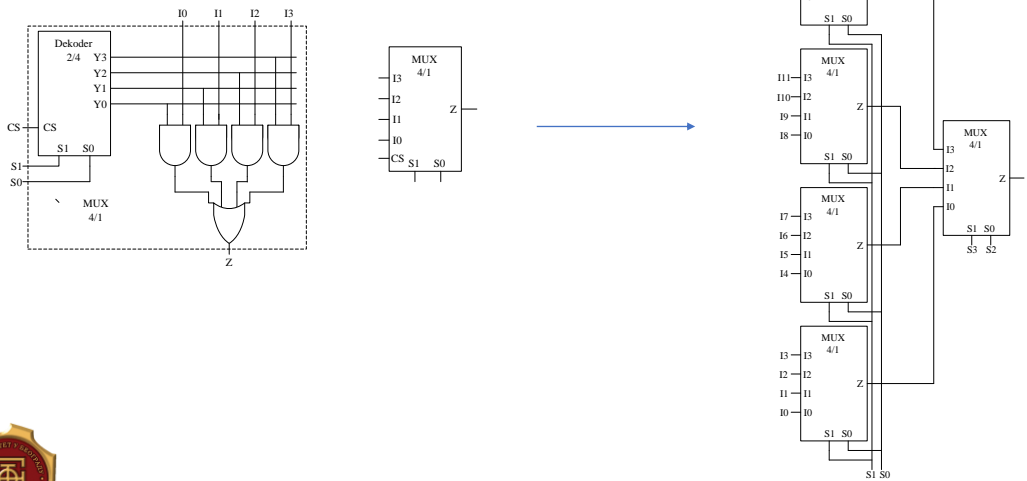


odnosno da ponovo imamo selekciju potpunih proizvoda koje formiraju signali S1 i S0. U tom smislu jedino što treba da uradimo jeste da na ulaze multipleksera za odgovarajući proizvod dovedemo željene vrednosti funkcije



14

Kao i kod dekoderskih mreža komponenta može imati selekcionu signal komponente



Koder prioriteta

Osim multipleksera koji predstavlja specijalnu vrstu kodera 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 kodera 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

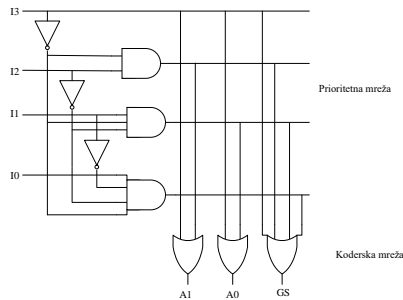
Neodređenost se pojavljuje u dve poslednje vrste. Preposlednja vrsta je situacija da je na ulazu sa najnižim indeksom aktivan signal, a poslednja da uopšte nema aktivnih signala. Iz tog razloga nam treba još jedan dodatni izlazni signal GS (group select) koji će pokazivati da li uopšte ima nešto da se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



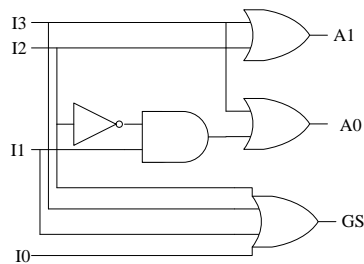
Proširena funkcionalna tabela u tom slučaju je

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“.



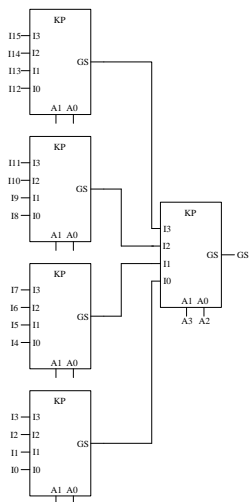
Prioritetna mreža će obezbediti da se u unutrašnjosti komponente pojavi samo jedna aktivna logička jedinica sa onog ulaza koji ima aktivnu jedinicu i najvišeg je prioriteta odnosno ima najveći indeks. U tom slučaju je lako napraviti Koderski deo mreže. Može i minimalnije da se nacrti



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



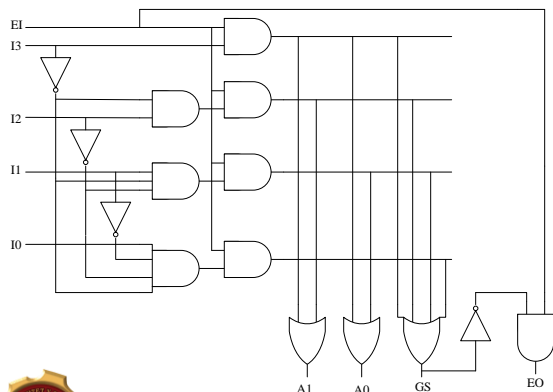
Prilikom pravljenja mreža većih kapaciteta, možemo se poslužiti idejama koje smo imali kod prethodnih kola. Na primer da je potrebno napraviti koder prioriteta sa 16 ulaza



Sa slike je vidljiva ideja da se iskoristi osobina koderu 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 koderu 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.



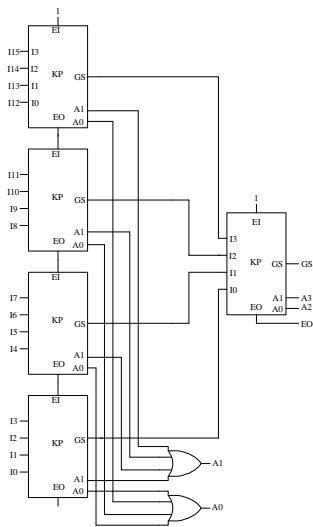
U tom smislu ćemo dodati ulazni signal EI (enable input) koji ako je aktivan, komponenta daje validne izlaze A1 i A0, a ako je neaktivan postavlja ih na logičke nule. Ovo možemo da uvedemo i za GS signal, u smislu da kada je on neaktivan komponenta nema šta da koduje bilo zato što nema aktivnih signala 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. 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 nivo 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



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.

