

# 4

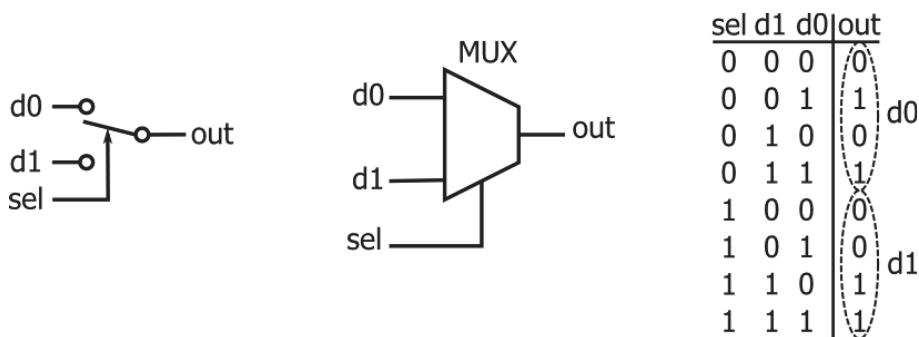
## Kombinacijski gradniki

Digitalna vezja in sisteme sestavljamo s povezavo vnaprej pripravljenih gradnikov, ki izvajajo različne logične funkcije. Gradniki so logična vezja, delimo jih na kombinacijska in sekvenčna. Kombinacijska vezja imajo izhod odvisen le od trenutne kombinacije na vhodu, pri sekvenčnih vezjih pa je odvisen še od shranjenega stanja. Obravnavali bomo kombinacijske izbiralnike, dekodirnike in osnovne pomnilne elemente.

### 4.1 Kombinacijski izbiralnik

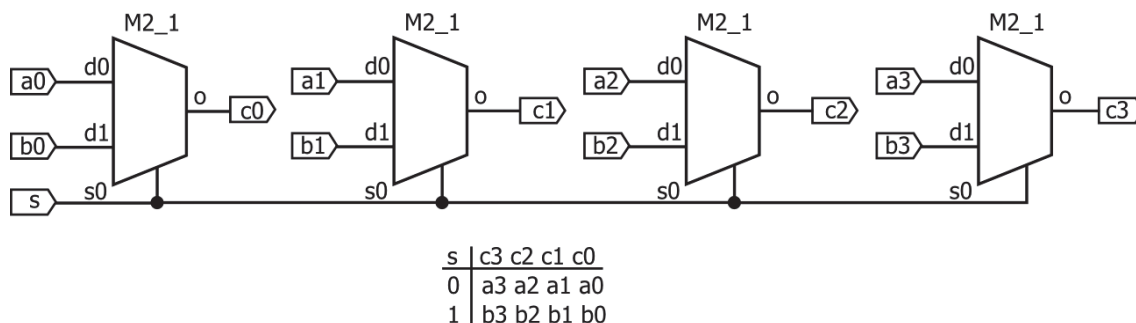


Kombinacijski izbiralnik ali multiplexer deluje kot preklopno stikalo, ki prenese na izhod vrednost enega izmed podatkovnih vhodov. Izbiralnik ima dva podatkovna vhoda, izbirni vhod in izhod. Ko je izbirni signal *sel* enak 0, dobimo na izhodu vrednosti z vhoda *d0*, sicer pa vrednosti z vhoda *d1*, kot prikazuje pravilnostna tabela.



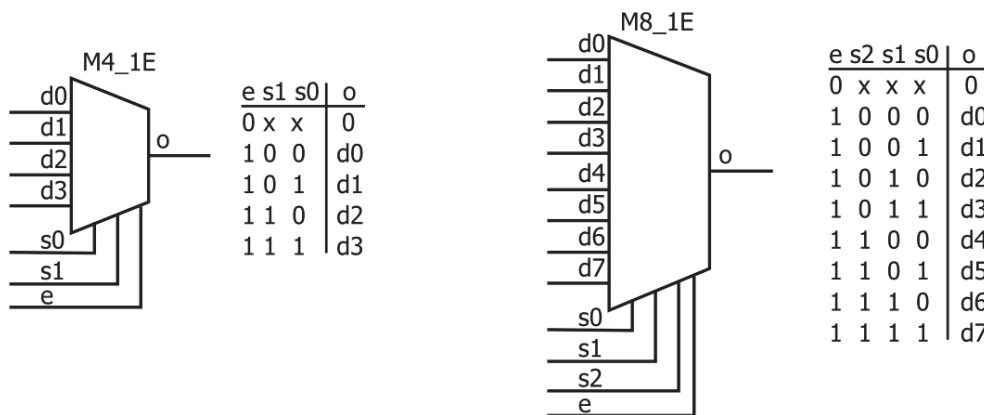
Slika 4.1: Izbiralnik 2-1: stikalna shema, simbol in pravilnostna tabela.

Podatkovni vhodi in izhod so lahko večbitne vrednosti. Izbiralnik z  $n$ -bitnimi podatkovnimi signali vsebuje  $n$  osnovnih izbiralnikov, pri katerih je izbirni vhod vezan skupaj. Oglejmo si vezje 4-bitnega izbiralnika 2-1, ki na izhod  $c$ , sestavljen iz signalov  $c_0$ – $c_3$  prenese enega izmed 4-bitnih vhodnih signalov  $a$  ali  $b$ .



Slika 4.2: Vezava štirih izbiralnikov v izbiralnik 4-bitnih vrednosti.

Naloga izbiralnika je prenašanje večjega števila vhodnih vrednosti prek enega izhodnega signala, kar izkoriščamo na primer v komunikacijskih vmesnikih. Nekateri izbiralniki imajo dodaten vhodni signal  $e$  (angl. enable) za omogočanje izhoda. Kadar je ta signal na logični 0, je izhod postavljen na 0, ne glede na stanje ostalih vhodov, kadar je signal  $e$  enak 1, pa deluje izbiralnik normalno.



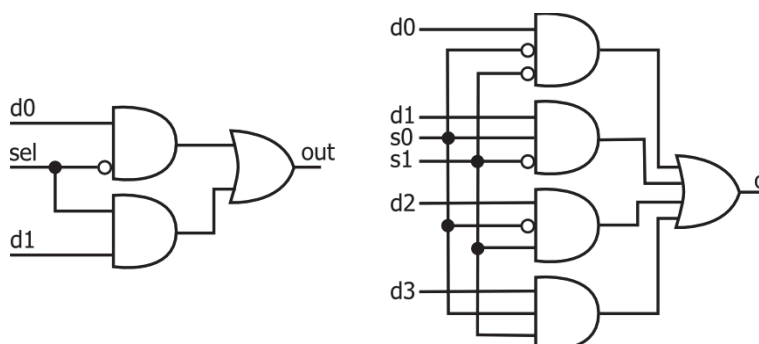
Slika 4.3: Izbiralnika 4-1 in 8-1 z dodatnim vhodom za omogočanje  $e$ .

V splošnem imajo izbiralniki več podatkovnih vhodov in večbitni izbirni vhod. Izbiralnik z oznako  $n-1$  ima  $n$ -bitni izbirni vhod in  $2^n$  podatkovnih vhodov. Slika 4.3 prikazuje simbole izbiralnikov 4-1 in 8-1. Podrobnosti delovanja teh izbiralnikov razberemo iz pravilnostnih tabel, ki so zapisane v zgoščeni obliki. Tabele ne vsebujejo vseh kombinacij vhodnih signalov, ker bi že za manjši izbiralnik 4-1 potrebovali kar 128 vrstic. Ko je vhodni signal  $e$  postavljen na 0, je izhod definiran ne glede na stanje ostalih vhodov, zato so njihova stanja označena z  $x$ .



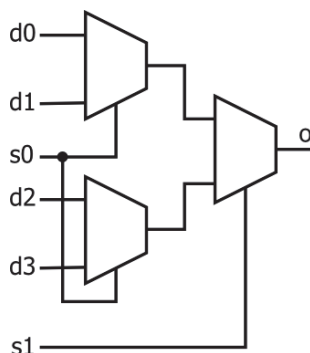
## 4.2 Izdelava izbiralnikov

Izbiralnik je kombinajsko vezje, ki ga lahko naredimo iz osnovnih logičnih vrat. Vezje sestavimo iz vrat AND, ki prepuščajo izbrani podatkovni vhod, in vrat OR, s katerimi združimo vse izhode. Izbirni signali gredo neposredno ali pa preko negatorjev na vrata AND. Kadar je kombinacija teh signalov na logični 1, bo izhod vrat enak podatkovnemu vhodu, v vseh ostalih primerih pa bo izhod enak 0. S primerno vezavo izbirnega signala zagotovimo, da bodo le ena logična vrata prepuščala izbrani vhodni signal.



Slika 4.4: Izvedba izbiralnikov 2-1 in 4-1 z logičnimi vrati.

Število podatkovnih vhodov izbiralnika lahko enostavno podvojimo z zaporedno vezavo treh izbiralnikov. Na sliki 4.5 je vezje izbiralnika 4-1, ki je narejeno iz treh manjših izbiralnikov 2-1.



Slika 4.5: Izvedba izbiralnika 4-1 z uporabo izbiralnikov 2-1.

Razlika med obema prikazanima izvedbama izbiralnika 4-1 je v vrsti in številu logičnih vrat: prva zahteva 4 trivhodna AND in ena OR, druga pa 6 dvovhodnih AND in 3 vrata OR. Pomembno je tudi število nivojev logičnih vrat med vhodom in izhodom: pri prvi izvedbi sta dva nivoja, pri drugi pa so štirje. Zaradi zakasnitev na štirih zaporednih logičnih vratih je drugo vezje počasnejše v primerjavi s prvim.

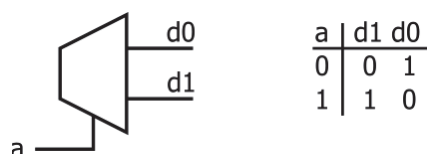
## 4.3 Kombinacijski dekodirnik



Z izrazom dekodirnik označujemo digitalna vezja, ki pretvarjajo eno obliko binarne kode v drugo. Med kombinacijske dekodirnike spadajo razdeljevalnik, binarni dekodirnik in splošno dekodirno vezje.

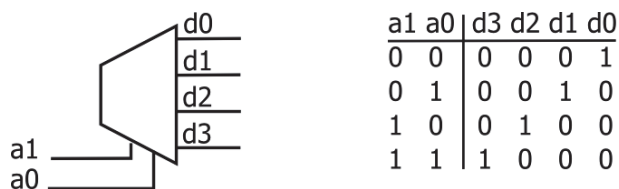
### Binarni dekodirnik in razdeljevalnik

Pri dvojiškem oz. binarnem dekodiranju pretvorimo  $n$  vhodnih signalov v  $2^n$  izhodnih. Izhodnih signalov je toliko, kolikor je dvojiških kombinacij na vseh vstopih: enovhodni dekodirnik ima 2 izhoda, 2-vhodni ima 4 izhode, 3-vhodni ima 8 izhodov, 4-vhodni pa 16 izhodov. Binarni dekodirnik ima le na enem izbranim izhodu logično 1, in sicer na tistem, ki ustreza trenutni vhodni kombinaciji.



Slika 4.6: Binarni dekodirnik 1-2: shematski simbol in pravilnostna tabela.

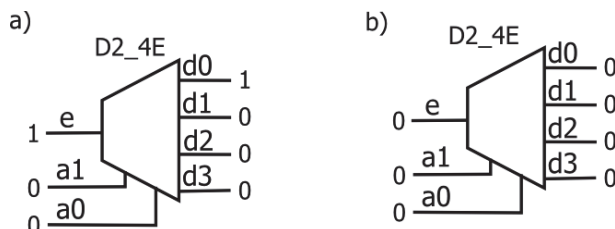
Slika 4.13 prikazuje grafični simbol dvobitnega dekodirnika in njegovo pravilnostno tabelo. Pri vhodni kombinaciji 0 0 je logična 1 na izhodu  $d_0$ , pri kombinaciji 0 1 je logična 1 na izhodu  $d_1$  itn.



Slika 4.7: 2-bitni binarni dekodirnik.

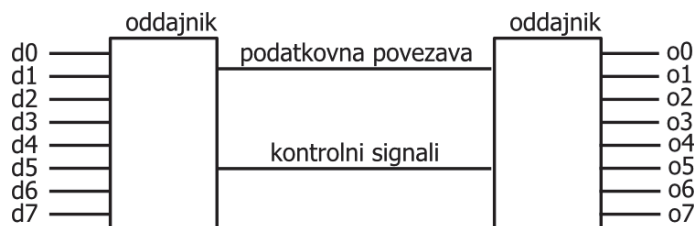
V digitalnem vezju uporabljamo binarni dekodirnik tam, kjer želimo s kombinacijo na vseh vstopih aktivirati natančno enega izmed množice gradnikov. Kombinacijo stanj na vstopih obravnavamo kot naslov ( $a$ , angl. address), ki je dodeljen izbranemu gradniku vezja. Gradnik v tem primeru služi kot dekodirnik naslovov.

Razdeljevalnik ali demultiplekser ima nasprotno funkcijo od izbiralnika, kar nakazuje tudi zrcaljen simbol. Razdeljevalnik temelji na binarnem dekodirniku z vhom za omogočanje. Signal za omogočanje obravnavamo kot podatkovni vhod, katerega vrednost se prenese na izbrani izhod. Kadar je signal  $e$  enak 1, dobimo na izbranem izhodu logično 1, kadar je  $e$  enak 0, so pa vsi izhodi na logični 0.



Slika 4.8: Razdeljevalnik a) pri  $e = 1$  deluje kot dekodirnik, b) pri  $e = 0$  so vsi izhodi 0.

Razdeljevalnik uporabljamo v komunikacijskih sistemih. Primer uporabe prikazuje slika 4.9, kjer z izbiralnikom združimo več vходов in jih prenesemo prek ene podatkovne povezave, razdeljevalnik pa opravi inverzno operacijo.



Slika 4.9: Vežje za kombiniranje (multipleksiranje) podatkovnih signalov.

## 7-segmentni dekodirnik

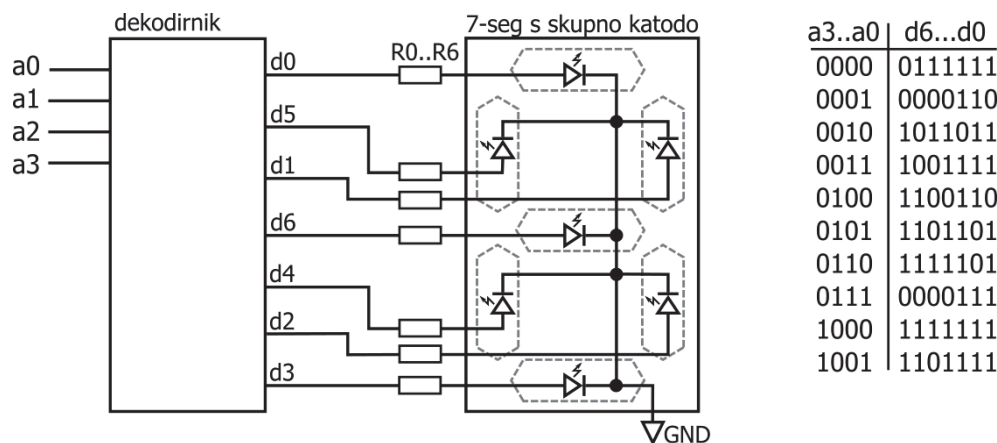
Poleg binarnih dekodirnikov poznamo še vrsto drugih dekodirnih vezij, ki imajo vsi skupno lastnost, da kombinacijo stanj na vhomih pretvorijo v neko drugo kombinacijo na izhodih. Med bolj znanimi je dekodirnik za 7-segmentne prikazovalnike. Prikazovalnik je sestavljen iz sedmih svetlečih diod, ki so razporejene tako, da lahko prikažejo desetiške številke (slika 4.10).



Slika 4.10: Prikaz desetiških števk na 7-segmentnem prikazovalniku.

Desetiške številke predstavimo v dvojiški obliki s 4-bitno vrednostjo. Dekodirnik pretvarja 4-bitne vhode  $a_0$  do  $a_3$  v 7-bitne izhode  $d_0$  do  $d_6$ , ki so vezani na posamezne segmente za prikaz

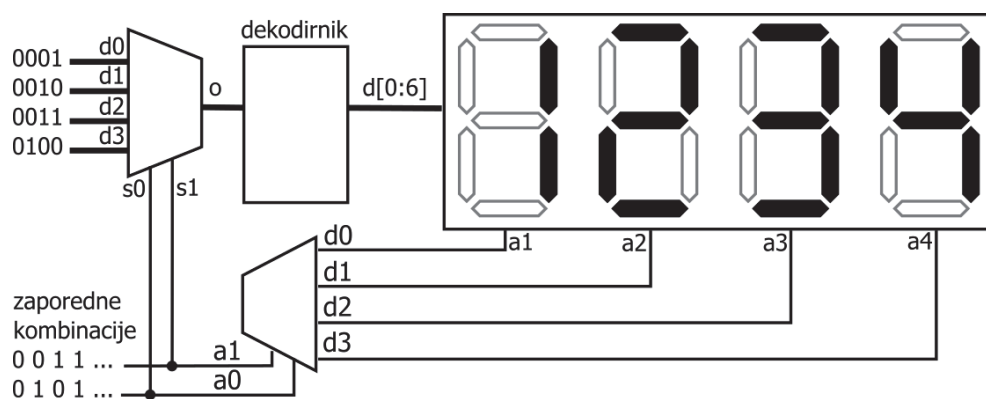
desetiških števk. Slika 4.11 prikazuje priklop dekodirnika na prikazovalnik in pravilnostno tabelo za prikaz števk med 0 in 9.



Slika 4.11: Vezava dekodirnika za 7-segmentni prikazovalnik in pravilnostna tabela.

Poznamo prikazovalnike LED s skupno katodo in prikazovalnike LED s skupno anodo. Nekateri prikazovalniki imajo še dodatno diodo za prikaz decimalne pike. Segmente prikazovalnikov s skupno katodo prižigamo z logično 1, katode vseh svetlečih diod pa morajo biti vezane na maso (slika 4.11).

Prikazovalniki s skupno anodo imajo skupni priključek vezan na  $V_{dd}$  in posamezne segmente prižigamo z logično 0. Ker delujejo LED z negativno logiko, je treba v pravilnostni tabeli dekodirnika negirati izhode. Takšen dekodirnik je narejen v integriranem vezju 74LS47 (shema na sliki 1.4).



Slika 4.12: Skupna vezava štirih LED prikazovalnikov s skupno anodo.

Kadar potrebujemo v vezju več prikazovalnikov, povežemo njihove podatkovne signale ( $d_0$  do  $d_6$ ) skupaj, s signali na skupnih anodah ali katodah pa izmenično prižigamo posamezne prikazovalnike. Če imamo štiri prikazovalnike, tako namesto  $4 \times 7 = 28$  signalov potrebujemo le 7 podatkovnih in 4 izbirne signale. Z dekodirnikom poskrbimo, da je v vsakem trenutku prižgan

le en prikazovalnik. Če izbrani prikazovalnik zamenjamo vsaj 20-krat v sekundi, naše oko ne bo opazilo menjav in bomo videli prižgane vse prikazovalnike hkrati. Na dekodirnik moramo pošiljati zaporedne binarne kombinacije, v vezju pa je še izbiralnik 4-1, ki poskrbi, da se v ritmu menjave izbranih prikazovalnikov spreminjajo tudi dekodirani podatkovni signali.

## 4.4 Izdelava dekodirnikov



Do izvedbe binarnega dekodirnika z osnovnimi logičnimi vrati pridemo tako, da zapišemo logične enačbe za posamezne signale. V dvobitnem dekodirniku je izhod  $d_0$  enak 1, kadar sta oba vhoda  $a_0$  in  $a_1$  enaka 0: vhoda torej negiramo in uporabimo operacijo AND. Podobno razmišljamo za ostale izhode in zapišemo enačbe:

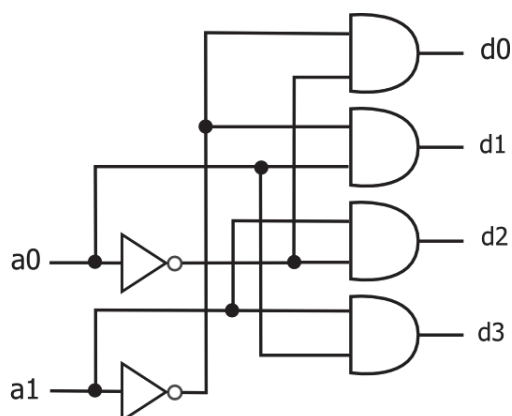
$$d_0 = \text{NOT}(a_0) \text{ AND } \text{NOT}(a_1)$$

$$d_1 = a_0 \text{ AND } \text{NOT}(a_1)$$

$$d_2 = \text{NOT}(a_0) \text{ AND } a_1$$

$$d_3 = a_0 \text{ AND } a_1$$

$$\begin{aligned} d_0 &= \text{NOT } a_0 \text{ AND } \text{NOT } a_1 \\ d_1 &= a_0 \text{ AND } \text{NOT } a_1 \\ d_2 &= \text{NOT } a_0 \text{ AND } a_1 \\ d_3 &= a_0 \text{ AND } a_1 \end{aligned}$$



Slika 4.13: Izvedba 2-bitnega dekodirnika z logičnimi vrati.

Dekodirnik je narejen iz štirih logičnih vrat AND in negatorjev. V logičnih izrazih opazimo, da se negirana signala  $a_0$  in  $a_1$  pojavljata večkrat, kar izkoristimo za optimizacijo vezja. Namesto da bi uporabili štiri negatorje, uporabimo le dva in povežemo negiran signal na več vrat AND, kot prikazuje slika 4.13.

Razdeljevalnik ima enako strukturo kot dekodirnik. Dodaten vhodni signal  $e$  pripeljemo na vsa logična vrata AND, saj predstavlja dodaten pogoj, da je na izbranem izhodu logična 1. V shemi dvobitnega razdeljevalnika bi tako imeli 3-vhodna logična vrata AND.

## 4.5 Načrtovanje vezij z izbiralniki

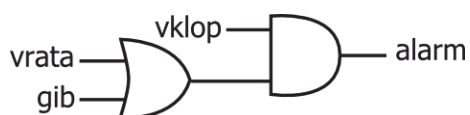


Kombinacijska vezja naredimo na podlagi pravilnostne tabele z osnovnimi logičnimi vrati ali pa z izbiralniki. V pravilnostni tabeli izbiralnika so na izhodu vsi podatkovni vhodi. Če povežemo na podatkovne vhode konstantne vrednosti 0 ali 1, lahko naredimo poljubno pravilnostno tabelo.

Za izvedbo kombinacijskega vezja z več izhodnimi signali uporabimo več izbiralnikov. V splošnem kombinacijskem vezju, brez optimizacije, potrebujemo toliko izbiralnikov, kot je izhodnih signalov. Velikost izbiralnikov je odvisna od števila vhodov kombinacijskega vezja.

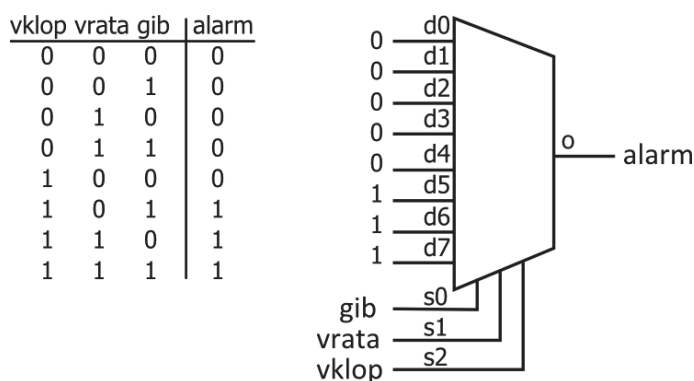
### Avtomobilski alarm z izbiralnikom

Z izbiralnikom naredimo vezje avtomobilskega alarma, ki se sproži, kadar je vklopljen in je aktiviran senzor gibanja ali pa so odprta vrata. Izvedbo alarma z logičnimi vrati smo si že ogledali:



Slika 4.14: Logično vezje za avtomobilski alarm.

Če želimo to vezje narediti z izbiralnikom, moramo najprej napisati pravilnostno tabelo za izhodni signal. Vezje ima 3 vhodne signale in v pravilnostni tabeli je  $2^3 = 8$  kombinacij. Kombinacijsko vezje avtomobilskega alarma naredimo z enim izbiralnikom 8-1, ki ima na podatkovnih vhodih konstante logične vrednosti, kot so določene v tabeli.

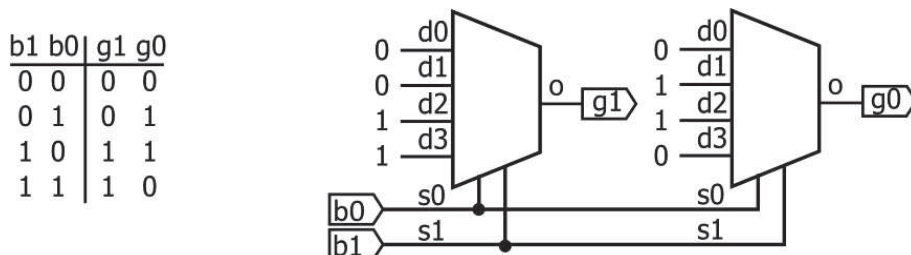


Slika 4.15: Izvedba avtomobilskega alarma z izbiralnikom.

### Grayev dekodirnik

Naredimo dvobitni dekodirnik, ki pretvarja vhodno dvojiško kodo v Grayevo kodo. Značilnost Grayeve kode je, da se pri zaporednih vrednostih vedno spremeni le en bit. Slika 4.16 prikazuje logično tabelo 2-bitne pretvorbe in izvedbo vezja z izbiralniki.



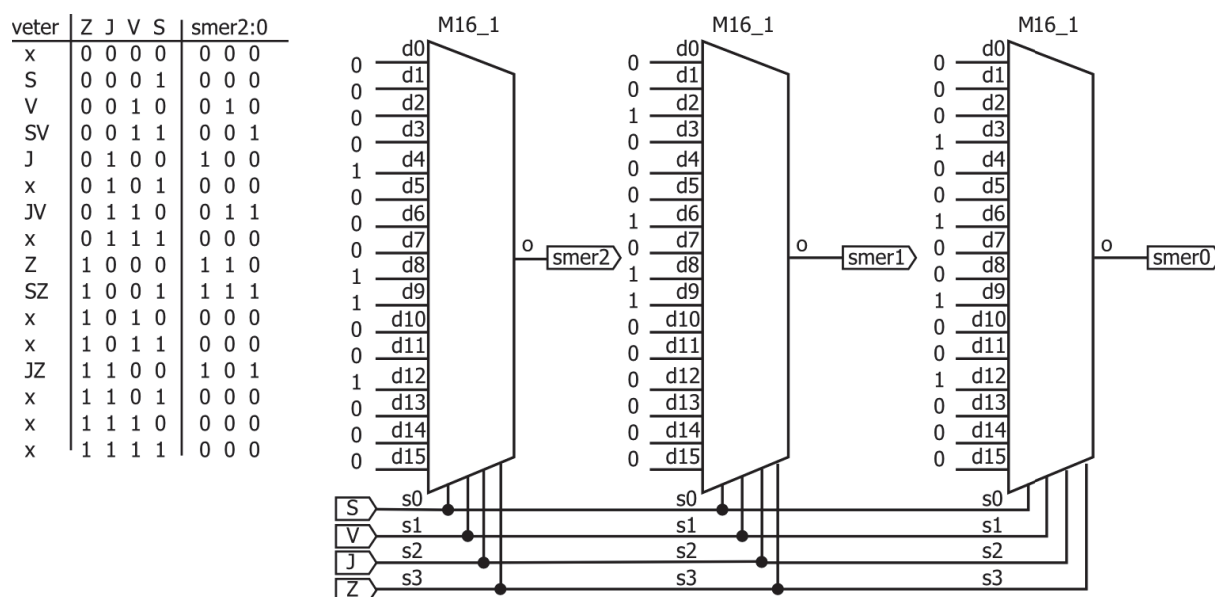


Slika 4.16: Grayev dekodirnik: pravilnostna tabela in izvedba z izbiralniki.

Dvobitni dekodirnik ima dva vhodna signala in dva izhodna signala. Za vsak izhod potrebujemo svoj izbiralnik 4-1, ki ima dva izbirna vhoda. Vezje lahko poenostavimo, če upoštevamo, da so vrednosti signala  $g1$  enake signalu  $b1$ , zato v praksi potrebujemo le en izbiralnik za določitev signala  $b0$ .

## Dekodirnik za smer vetra

Naredimo vezje za dekodiranje smeri vetra. Iz vetrnice dobimo 4 signale:  $S$ ,  $J$ ,  $V$  in  $Z$ , med katerimi sta eden ali dva postavljena na 1, ostali pa so 0. Naloga dekodirnika je pretvoriti 4-bitni vhod v 3-bitno kombinacijo, ki jo bomo določili s pravilnostno tabelo. Tabela naj definira vse kombinacije, tudi tiste, ki jih ne pričakujemo (označene z x):

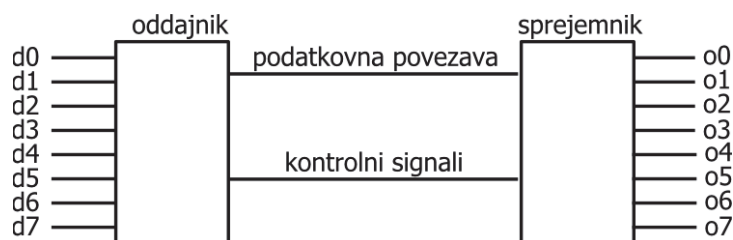


Slika 4.17: Dekodirnik smeri vetra z izbiralniki.

V vezju uporabimo tri izbiralnike 16-1, ki imajo izbirne vhode  $s0$  do  $s3$  vezane skupaj in priključene na vhodne signale. Na podatkovne vhode povežemo konstante iz pravilnostne tabele. Pri tem moramo paziti na vrstni red: izbirni signal  $s3$  je vezan na signal na levi strani pravilnostne tabele,  $s2$  na naslednji signal itn.

## Naloge

1. Razmisli, kako bi sestavil izbiralnik 8-1, če imaš na voljo le izbiralnike 4-1 in 2-1. Nariši shemo vezave!
2. Opiši zgradbo in delovanje vezja za prenos osmih signalov preko ene podatkovne povezave. Katere kombinacijske gradnike potrebuješ na oddajni in sprejemni strani?



3. Ugotovi, koliko izbiralnikov in kakšne vrste izbiralnike potrebujemo za izdelavo 7-segmentnega dekodirnika.

