

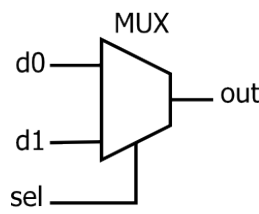
4

Kombinacijski gradniki

Digitalna vezja in sisteme sestavljamo s povezavo vnaprej pripravljenih gradnikov, ki izvajajo različne funkcije. Gradniki so sestavljeni iz osnovnih logičnih vrat in imajo običajno več vhodov in izhodov. V poglavju bomo spoznali *kombinacijske* logične gradnike, pri katerih je vrednost na izhodu odvisna od kombinacije vrednosti na vhodih.

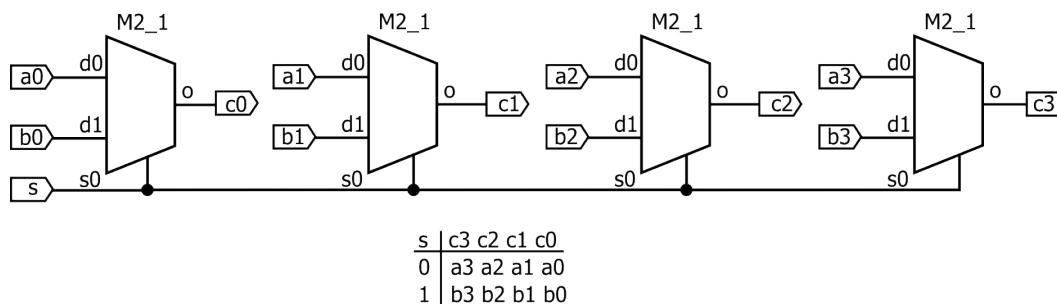
4.1 Izbiralnik (multiplekser)

Kombinacijski izbiralnik ali multiplekser je vezje, ki glede na stanje izbirnega vhoda prenese vrednost enega izmed podatkovnih vhodov na izhod. Slika 4.1 prikazuje najbolj osnovno izvedbo izbiralnika 2-1 z dvema podatkovnima vhodoma, enobitnim izbirnim vhodom in enim izhodom. Ko je izbirni signal *sel* enak 0, dobimo na izhodu vrednosti iz vhoda *d0*, sicer pa vrednosti iz vhoda *d1*, kot prikazuje pravilnostna tabela.

			
sel	d1	d0	out
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

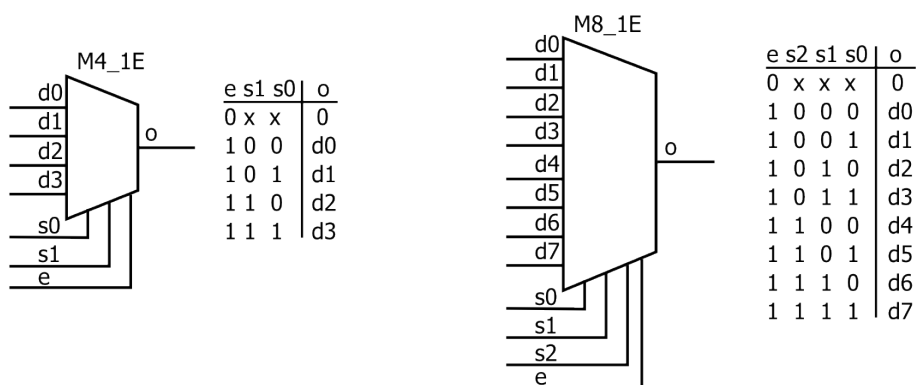
Slika 4.1: Simbol izbiralnika in pravilnostna tabela.

Podatkovni vhodi in izhod so lahko večbitne vrednosti. Izbiralnik z k -bitnimi podatkovnimi signali je narejen iz k osnovnih izbiralnikov pri katerih izbirni vhod večemo skupaj. Primer izbiralnika 2-1, ki na 4-bitni izhod c prenese enega izmed 4-bitnih vhodnih signalov a ali b je prikazan na sliki 4.2.



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

Naloga izbiralnika je prenašanje večjega števila vhodnih vrednosti preko enega izhodnega signala, kar izkoriščamo naprimer 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.

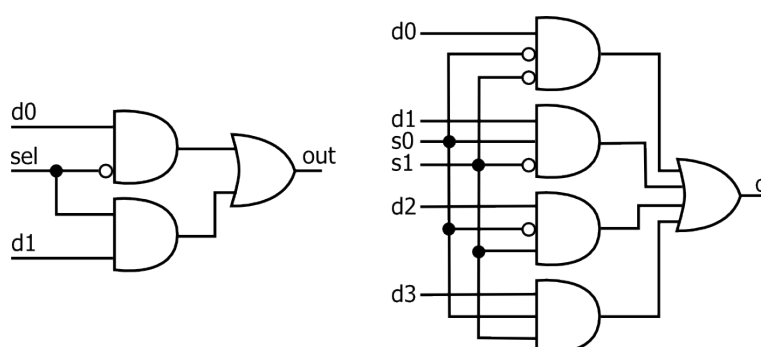


Slika 4.3: Izbiralnika 4-1 in 8-1 z dodatnim vhodom za omogočanje e .

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 zgoščenih oblikah. 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 .

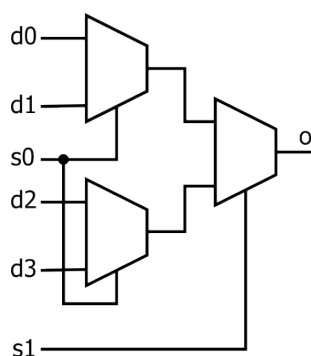
4.1.1 Izdelava izbiralnikov

Izbiralnik je kombinacijsko 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 prikazuje izvedbo izbiralnikov 2-1 in 4-1 z logičnimi vrati.



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, kot prikazuje slika 4.5. Na sliki je narejen izbiralnik 4-1 z uporabo treh manjših izbiralnikov 2-1.



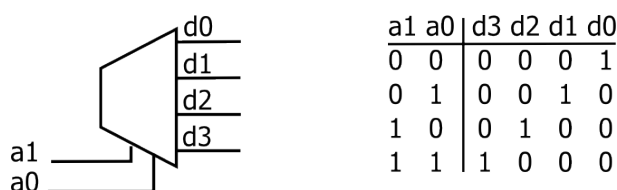
Slika 4.5: Izvedba izbiralnika 4-1 z uporabo izbiralnikov 2-1.

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.

4.2 Razdeljevalnik in binarni dekodirnik

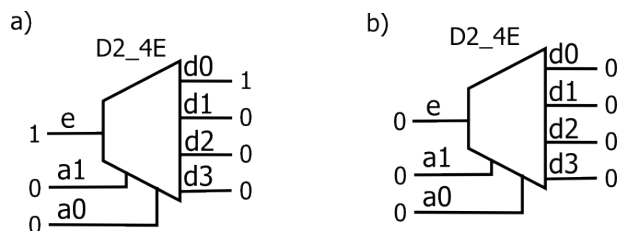
Razdeljevalnik ali demultiplekser ima nasprotno funkcijo od izbiralnika. Razdeljevalnik ima n izbirnih vhodov in 2^n izhodov: 2-bitni razdeljevalnik ima 2 vhoda in 4 izhode, 3-bitni ima 3 vhode in 8 izhodov, 4-bitni pa 4 vhode in 16 izhodov.

Slika 4.8 prikazuje grafični simbol dvobitnega razdeljevalnika in njegovo pravilnostno tabelo. Glede na kombinacijo na izbirnih vseh $a1$ in $a0$ se postavi eden izmed izhodov $d0$, $d1$, $d2$ ali $d3$ na vrednost 1, vsi ostali pa so 0. S kombinacijo 0 0 izberemo $d0$, z 0 1 izberemo $d1$, itn.



Slika 4.6: 2-bitni razdeljevalnik.

Podobno kot pri izbiralniku ima lahko tudi razdeljevalnik dodaten vhod 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, kot prikazuje slika 4.7. Takšen razdeljevalnik imenujemo tudi *Binarni dekodirnik*.



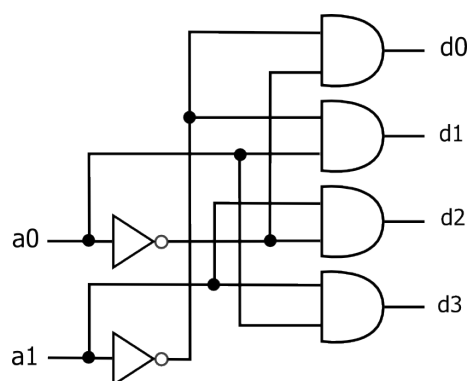
Slika 4.7: Binarni dekodirnik a) pri $e=1$ deluje normalno, b) pri $e=0$ so vsi izhodi 0.

4.2.1 Izvedba z logičnimi vrati

Do izvedbe razdeljevalnika z osnovnimi logičnimi vrati pridemo tako, da zapišemo logične enačbe za posamezne signale. V dvobitnem razdeljevalniku je izhod $d0$ enak 1, kadar sta oba vhoda $a0$ in $a1$ enaka 0: vhoda torej negiramo in uporabimo operacijo AND. Podobno razmišljamo za ostale izhode in zapišemo enačbe:

$$\begin{aligned}
 d0 &= \text{NOT}(a0) \text{ AND } \text{NOT}(a1) \\
 d1 &= a0 \text{ AND } \text{NOT}(a1) \\
 d2 &= \text{NOT}(a0) \text{ AND } a1 \\
 d3 &= a0 \text{ AND } a1
 \end{aligned}$$

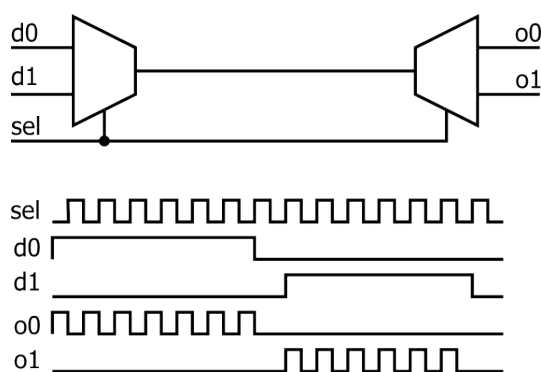
Razdeljevalnik je narejen iz štirih logičnih vrat AND in negatorjev. V logičnih izrazih opazimo, da se negirana signala $a0$ in $a1$ pojavljata večkrat, kar izkoristimo za optimizacijo vezja. Namesto da bi uporabili štiri negatorje, uporabimo le dva in povežemo negiran signal na več vrat AND, kot prikazuje slika.



Slika 4.8: Izvedba 2-bitnega razdeljevalnika z logičnimi vrati.

4.2.2 Uporaba razdeljevalnikov in dekodirnikov

V digitalnem vezju uporabljamo razdeljevalnik tam, kjer želimo s kombinacijo na izbirnih vseh aktivirati natančno enega izmed množice gradnikov. Kombinacijo stanj na izbirnih vseh obravnavamo kot celoštevilski naslov, ki je dodeljen izbranemu gradniku vezja. Razdeljevalnik v tem primeru služi kot dekodirnik naslovov.

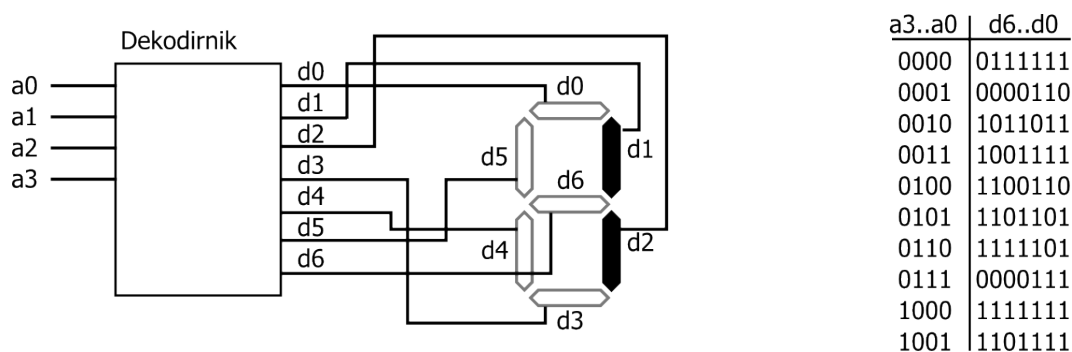


Slika 4.9: Časovno kombiniranje (multipleksiranje) podatkovnih signalov.

Binarni dekodirnik uporabljamo v komunikacijskih sistemih. Primer uporabe prikazuje slika 4.9, kjer z izbiralnikom združimo dva podatkovna kanala v enega, binarni dekodirnik pa opravi inverzno operacijo. Kadar želimo sočasno kombinirati oz. multipleksirati prenos podatkovnih kanalov, je potrebno izbirni signal preklapljeti z dovolj visoko frekvenco, kakor prikazuje časovni diagram.

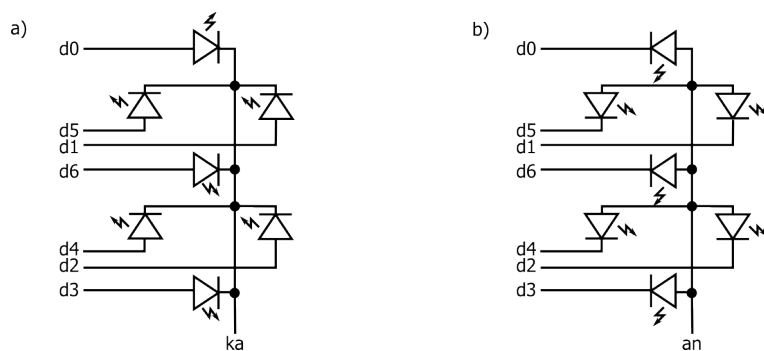
4.3 Kombinacijski dekodirnik

Poleg binarnih dekodirnikov poznamo še vrsto drugih dekodirnih vezij, ki imajo vsi skupno lastnost, da kombinacijo stanj na vseh vhodih pretvorijo v neko drugo kombinacijo na izhodih. Dekodirniki spadajo po definiciji med kombinacijska vezja. Med bolj znanimi je dekodirnik za 7-segmentne prikazovalnike, ki pretvarja 4-bitno število v kombinacijo za prikaz desetiške številke (slika 4.10). Prikazovalnik je sestavljen iz sedmih svetlečih diod, ki so razporejene tako, da lahko prikažejo desetiške številke.



Slika 4.10: Vezava dekodirnika za 7-segmentni prikazovalnik in pravilnostna tabela.

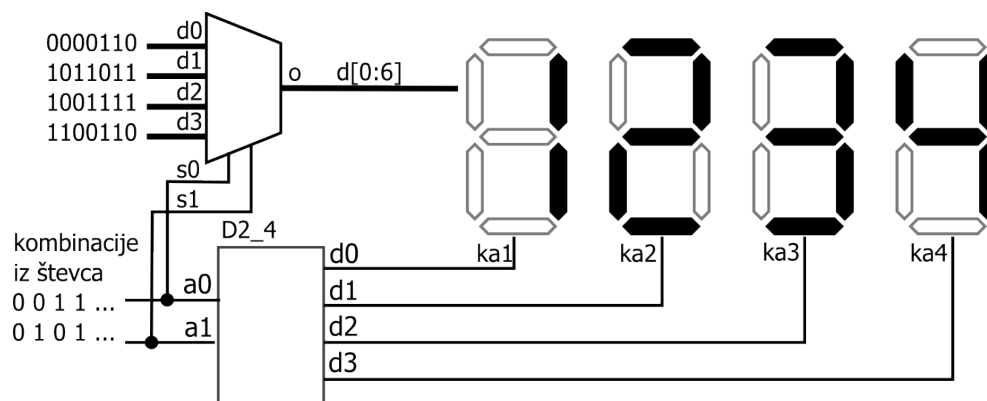
Poznamo dve vrsti LED prikazovalnikov: s skupno katodo in s skupno anodo, kot prikazuje slika 4.11. Segmente prikazovalnikov s skupno katodo prižigamo z logično 1, katodo pa vezemo preko upora na maso. Prikazovalnik s skupno anodo pa ima skupni priključek vezan na Vdd in posamezne segmente prižigamo z logično 0. V tem primeru moramo izhode dekodirnika negirati. Nekateri prikazovalniki imajo še dodatno diodo (signal $d7$) za prikaz decimalne pike.



Slika 4.11: LED prikazovalniki: a) s skupno katodo in b) s skupno anodo.

Kadar imamo v vezju več 7-segmentnih prikazovalnikov, so običajno podatkovni signali ($d0$ do $d6$) vezani 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 logičnim vezjem poskrbimo, da je v vsakem trenutku prižgan le en prikazovalnik. Če izbrani prikazovalnik dovolj hitro menjavamo, npr. vsaj $20 \times$

sekundi, naše oko ne bo opazilo zamenjav in bomo videli kot da so prižgani vsi prikazovalniki skupaj. Izbrani prikazovalnik določimo s števcem, ki poskrbi za zaporedne binarne kombinacije in binarnim dekodirnikom, kot prikazuje shema na sliki 4.12. V vezju je uporabljen še izbiralnik 4-1, ki poskrbi, da se v ritmu menjave izbranih prikazovalnikov spreminjajo tudi podatkovni signali.



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

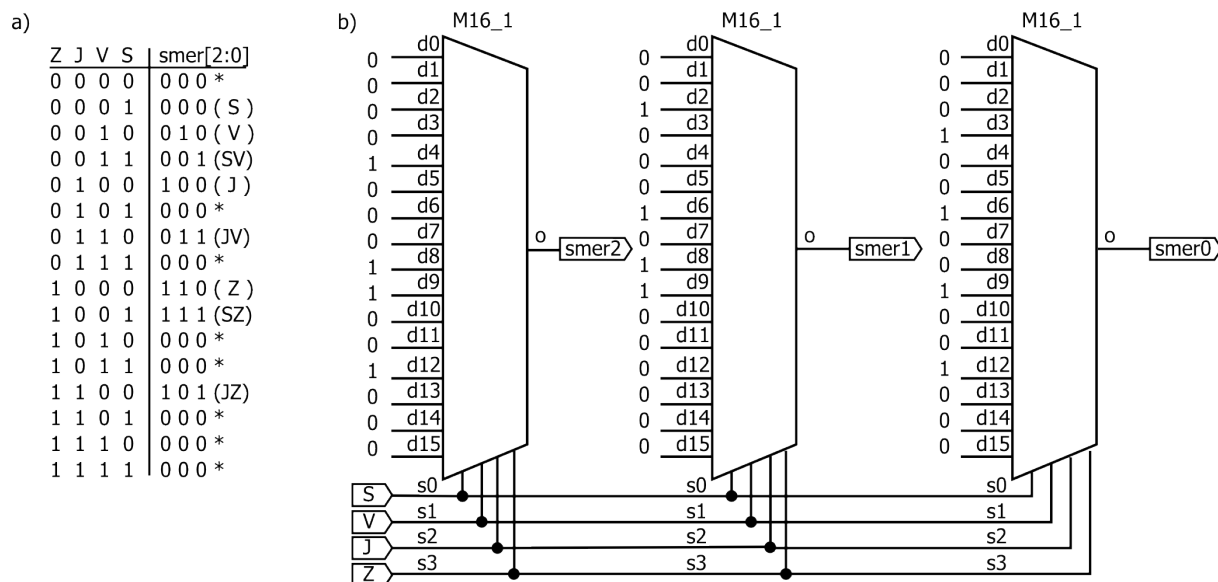
4.3.1 Izvedba dekodirnikov

Dekodirnike naredimo na podlagi pravilnostne tabele z osnovnimi logičnimi vrati ali pa z izbiralniki. Za izvedbo potrebujemo toliko izbiralnikov, kot je izhodnih signalov, velikost izbiralnikov pa je odvisna od števila vhodov dekodirnika.

Vzemimo primer vezja za dekodiranje smeri vetra. Iz senzorja (vetrnice) dobimo 4 signale: S , J , V in Z , med katerimi je eden ali dva postavljen na 1, ostali pa so 0. Naloga dekodirnika je pretvoriti 4-bitni vhod v 3-bitno kombinacijo, ki kodira smer vetra po pravilu:

- $smer = "000"$ kadar je $S = 1$,
- $smer = "001"$ kadar je $S = 1$ in $V = 1$,
- $smer = "010"$ kadar je $V = 1$,
- $smer = "011"$ kadar je $J = 1$ in $V = 1$,
- $smer = "100"$ kadar je $J = 1$,
- $smer = "101"$ kadar je $J = 1$ in $Z = 1$,
- $smer = "110"$ kadar je $Z = 1$,
- $smer = "111"$ kadar je $S = 1$ in $Z = 1$.

Iz opisa naredimo pravilnostno tabelo z vsemi kombinacijami štirih vhodov, pri kateri določimo kakšen naj bo izhod tudi v primerih, ko je več vhodov postavljenih na 1. Na sliki 4.13 a) je pravilnostna tabela dekodirnika smeri vetra v kateri so z zvezdico označene pričakovane kombinacije vhodov.



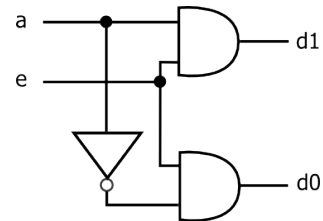
Slika 4.13: Dekodirnik smeri vetra: a) pravilnostna tabela in b) izvedba z izbiralniki.

Kodirnik smeri vetra ima $2^4 = 16$ kombinacij in za vsak bit izhoda uporabimo en izbiralnik 16-1, kot prikazuje slika 4.13 b). Na podatkovne vhode postavimo konstantne vrednosti, ki predstavljajo posamezen stolpec pravilnostne tabele. Izbirni vhodi s_0 do s_3 obeh izbiralnikov so vezani skupaj in priključeni na vhodne signale. Pri tem moramo paziti na vrstni red: izbirni signal s_3 je vezan na signal na levi strani pravilnostne tabele, s_2 na naslednji signal itn.

4.4 Naloge

1. Kakšen bo izhod vezja, ko je na vhodu $a = 1, e = 1$?

- (a) $d0 = 0, d1 = 0$
- (b) $d0 = 1, d1 = 0$
- (c) $d0 = 0, d1 = 1$
- (d) $d0 = 1, d1 = 1$



2. Na podatkovnih vhodih izbiralnika so konstantne vrednosti, kot prikazuje slika. Pri kateri kombinaciji kontrolnih vhodov bo izhod enak 1?

- (a) $s0 = 0, s1 = 0, e = 0$
- (b) $s0 = 0, s1 = 0, e = 1$
- (c) $s0 = 0, s1 = 1, e = 1$
- (d) $s0 = 1, s1 = 0, e = 1$

