

2

Signali in logična vrata

Digitalni signali so signali, ki lahko zavzamejo le končno število različnih stanj. Imenujemo jih tudi *diskretni signali*. Preprost primer digitalnega signala je število dvignjenih prstov na roki - število je lahko le ena izmed vrednosti iz diskretnega območja med 0 in 10. Beseda *digitalni* prihaja iz latinskega izraza digitus, ki pomeni prst. V tem poglavju bomo predstavili digitalne signale v obliki dvojiškega zapisa in napetostnih nivojev v elektronskem vezju.

2.1 Binarni signali



V digitalnih elektronskih vezjih se največkrat uporabljajo *binarni* signali, ki lahko zavzemajo le dve možni stanji označeni kot:

- napačno (false) ali pravilno (true),
- nizko (potencial V_L) ali visoko (V_H),
- številka 0 ali 1.

Dve stanji opisujeta preproste pojave, kot so prižgana oziroma ugasnjena žarnica ali stikalo. Predstavljata lahko logično trditev, ki je pravilna ali napačna. Zapis stanj v obliki pravilno (true) ali napačno (false) je primeren za obravnavo vezij, ki izvajajo odločitvene operacije. Primer preproste logične operacije je negacija: napačno stanje spremenimo v pravilno in obratno. Element, ki izvaja logično negacijo imenujemo logični negator ali inverter.

Najbolj pogost zapis digitalnih stanj je v obliki številskih vrednosti 0 ali 1. Takšen zapis ni samo najkrajši, ampak je tudi primeren za računanje, saj digitalna vezja velikokrat izvajajo računske operacije. Vrednost 0 ali 1, ki jo zavzame enostaven signal, imenujemo binarna številka

ali *bit* (angl. binary digit). Vodila v digitalnih vezjih pa prenašajo večbitne vrednosti v dvojiškem zapisu.

Za razumevanje dvojiških vrednosti si najprej pogledimo kako so sestavljena večmestna desetiška števila. Vrednost desetiškega števila lahko zapišemo kot vsoto števk pomnoženih s koeficienti potence 10. Enice množimo z 10^0 , desetice z 10^1 , stotice z 10^2 itn. Primer:

$$523_{(10)} = 5 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$$

Popolnoma enako velja za dvojiška števila, le da pri izračunu desetiške vrednosti uporabimo potence števila 2. Posamezne števice dvojiškega števila pomnožimo s potencami števila 2 in seštejemo. Pri zapisu celega števila množimo skrajno desno dvojiško števko z 2^0 , naslednjo z 2^1 in tako dalje. Pogledjmo si primer izračuna desetiške vrednosti 4-bitnega števila:

$$1100_{(2)} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8 + 4 = 12_{(10)}$$

Pretvorbo najlažje naredimo tako, da nad vsako števko zapišemo ustrezno potenco števila 2 in seštejemo tiste potence, pod katerimi je binarna števka 1:

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \\ \hline 1 & 1 & 0 & 0 \end{array}_{(2)} = 8 + 4 = 12$$

Prvi digitalni mikroprocesor Intel 4004 je računal s 4-bitnimi vrednostmi, ki v desetiškem sistemu pokrijejo območje le ene desetiške števice. Če bi želeli računati z dvomestnimi desetiškimi števili (vrednosti med 0 in 99), bi potrebovali 7-bitni dvojiški zapis z območjem med 0 in 127. Pri delu z dvojiškimi signali, je koristno če na pamet poznamo potence števila 2, ki določajo zgornjo mejo območja vrednosti, kot prikazuje tabela 2.1.

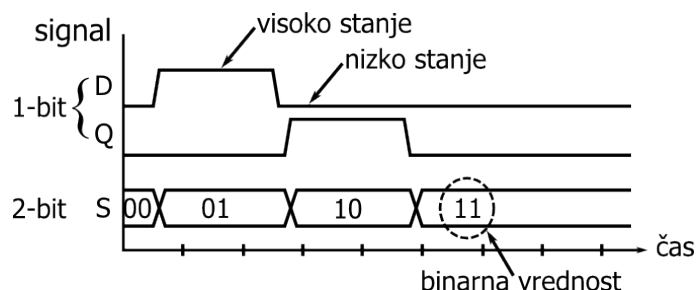
N	N-bitna števila	2^N	območje vrednosti
2	00 - 11	4	0 - 3
3	000 - 111	8	0 - 7
4	0000 - 1111	16	0 - 15
5	00000 - 11111	32	0 - 31
6	000000 - 111111	64	0 - 63
7	0000000 - 1111111	128	0 - 127
8	00000000 - 11111111	256	0 - 255
9	000000000 - 111111111	512	0 - 511
10	0000000000 - 1111111111	1024	0 - 1023

Tabela 2.1: Območja vrednosti binarnih pozitivnih števil

Tabela prikazuje območja vrednosti 2- do 10-bitnih binarnih števil. Obseg vrednosti N-bitnega števila izračunamo s potenco 2^N . Do sedaj smo se ukvarjali le z naravnimi števili, ki so samo pozitivna. Če želimo predstaviti cela števila, ki so pozitivna in negativna, moramo dodati še en bit za predznak.

Potek signalov, ki se spreminjajo s časom, opazujemo na časovnem diagramu (angl. waveform). Običajno opazujemo več signalov, ki jih vrišemo v en diagram z enako časovno skalo za

vse signale, kot prikazuje slika 2.1. Za opazovanje signalov, ki v vezju zelo hitro spreminjajo stanja, uporabimo ustrezen merilni inštrument: osciloskop ali logični analizator.



Slika 2.1: Časovni diagram digitalnih signalov

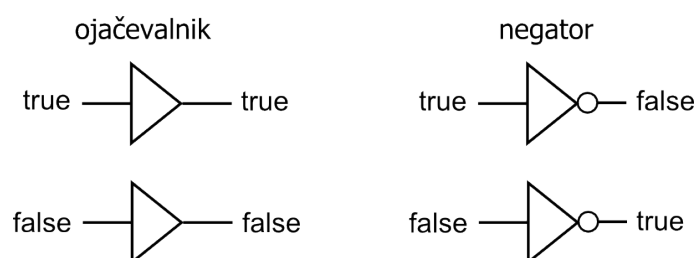
Večbitne signale predstavimo v časovnem diagramu kot en signal v katerem so zapisane trenutne vrednosti celotnega vodila. Programska oprema za prikazovanje časovnega diagrama omogoča nastavitve prikaza v binarni obliki ali pa dekodirani decimalni, šestnajstiški, ASCII ipd. Dvojiški zapis pretvorimo v šestnajstiški tako, da združujemo in pretvarjamo po 4 števke hkrati. Primer pretvorbe 8-bitne vrednosti:

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array} \quad \begin{array}{cccc} 8 & 4 & 2 & 1 \\ \hline 1 & 1 & 0 & 0 \end{array}_{(2)} = 1_{(10)} \quad 12_{(10)} = 1C_{(16)}$$

2.1.1 Logični napetostni nivoji



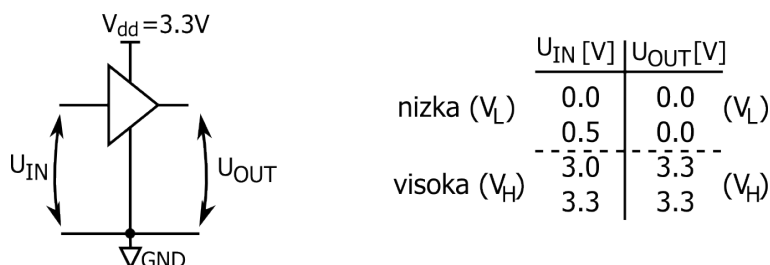
Gradniki digitalnih vezij imajo vhodne priključke, na katerih zaznavajo stanje binarnih signalov in izhodne priključke, na katere vsiljujejo binarne vrednosti. Za primer vzemimo dva najbolj enostavna gradnika, ki imata le en vhod in izhod. Slika 2.2 prikazuje delovanje logičnega ojačevalnika in logičnega negatorja. Ojačevalnik ima na izhodu enako logično stanje, kot je na vhodu, pri negatorju pa je stanje na izhodu obrnjeno.



Slika 2.2: Logični ojačevalnik in logični negator

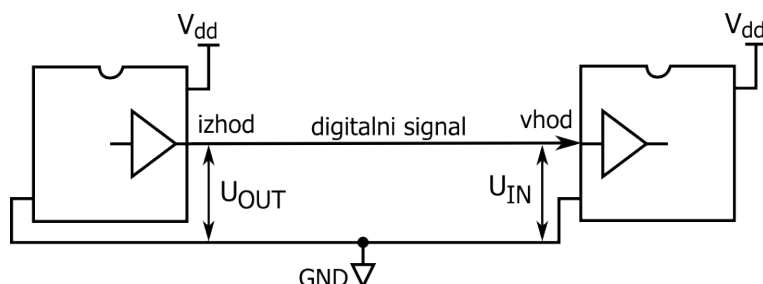
V elektronskem vezju predstavlja logično stanje potencial na priključku oziroma napetost priključka proti masi. Stanje logične ničle je določeno z nizkim (V_L) potencialom, stanje logične enice pa z visokim (V_H) potencialom signala oziroma nizko in visoko napetostjo proti masi.

Slika 2.3 prikazuje napetosti na logičnem ojačevalniku pri nizkem in visokem stanju na vhodu. Iz priložene tabele vidimo, da predstavljajo različne napetosti enako logično stanje, zato potrebujemo dogovor o logičnih napetostnih nivojih.



Slika 2.3: Potenciali in napetosti na vhodu in izhodu ojačevalnika

Osnovni gradniki digitalnih vezij se obnašajo kot preprosta elektronska stikala, ki preklaplajo med potencialom V_L in V_H . Elektronska stikala so bila včasih narejena z releji ali elektronkami, danes pa z različnimi elementi v polprevodniški tehnologiji. Z razvojem elektronike se spreminjajo tudi osnovni elementi in njihove električne lastnosti. Da bi digitalna vezja v različnih tehnologijah lahko povezali med seboj, moramo uvesti nek dogovor, ki določa potenciala za nizko in visoko stanje na vhodih in izhodih vezja. Vrednosti potencialov oz. napetosti gledamo v ustaljenem stanju, zato se dogovor imenuje statični red (angl. static discipline).



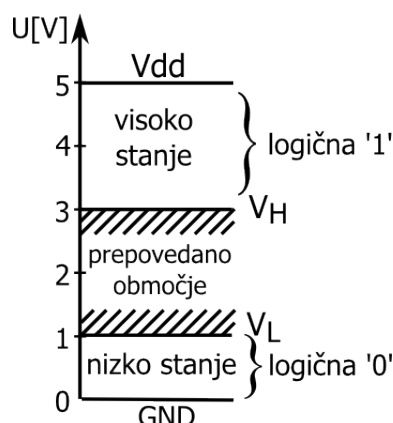
Slika 2.4: Povezava dveh digitalnih vezij

Elektronska stikala v digitalnih vezjih niso idealna, imajo neko upornost, ki povzroči da visoko stanje V_H ni enako napajalnemu potencialu V_{dd} in da je nizko stanje V_L nekoliko višje od potenciala mase. Prav tako moramo upoštevati možne razlike v napajalnih napetostih integriranih vezij. Statični red omogoča pravilno interpretacijo signalov, ki potujejo med dvema digitalnima vezjema. Vzemimo najbolj preprost in pogost primer digitalne povezave, ko je izhod enega vezja vezan na vhod drugega, kot prikazuje slika 2.4. Enostaven dogovor bi lahko določal, da predstavljajo vse napetosti med V_{dd} in $V_{dd}/2$ visoko stanje (logično '1'), napetosti med 0 in $V_{dd}/2$ pa nizko stanje (logično '0'):

$$\text{logična '0': } 0V \leq V_L \leq V_{dd}/2$$

$$\text{logična '1': } V_{dd}/2 \leq V_H \leq V_{dd}$$

Pri takšnem dogovoru se pojavi težava, če dobi sprejemnik na vhod napetost $V_{dd} / 2$. Da bi lahko sprejemnik nedvoumno razločeval med logično '0' in '1' dodamo prepovedano območje potencialov na signalni liniji, kot prikazuje slika 2.5. Vpeljali smo dva nova potenciala: V_{IH} je minimalni potencial, ki se na vhodu logičnega vezja interpretira kot visoko stanje, V_{IL} pa maksimalni, ki predstavlja nizko stanje.



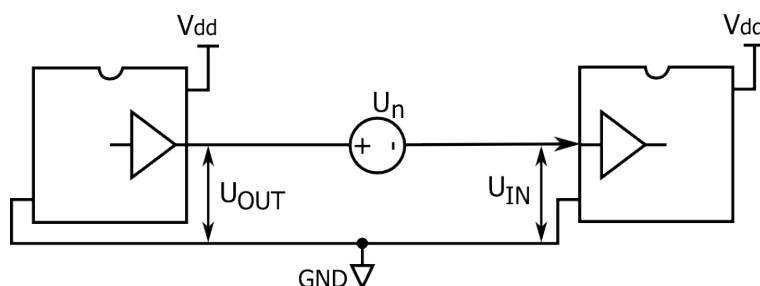
Slika 2.5: Dogovor o potencialih za nizko in visoko stanje

Konkretni vrednosti so odvisne od tehnologije in zahtev - večje prepovedano območje poveča robustnost sistema, večje območje pravičnih stanj pa združljivost z več tehnologijami. Za neko 5V CMOS tehnologijo, bi lahko uporabili vrednosti:

$$\text{logična '0': } 0V \leq V_L \leq 1V$$

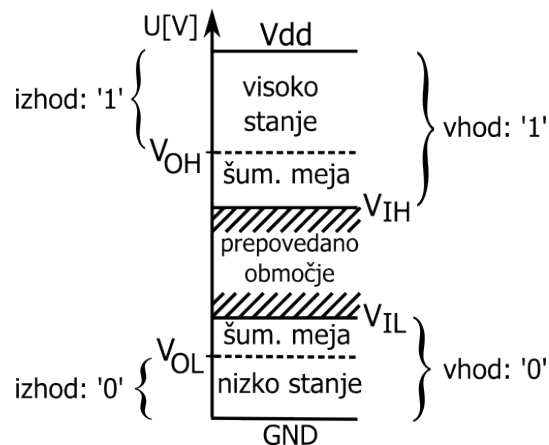
$$\text{logična '1': } 3V \leq V_H \leq 5V$$

Na signalni povezavi se lahko inducira šum, ki ga modeliramo kot dodatno napetost U_n na signalni povezavi. Šum lahko povzroči neveljavno stanje na vhodu sprejemnika, kljub temu da je na oddajni strani stanje z veljavnim potencialom. Inducirani šum je v splošnem pozitivna ali negativna napetost. Denimo, da se na povezavi inducira $U_n = 200 \text{ mV}$ šumne napetosti. Če je na izhodu CMOS vezja nizko stanje z napetostjo 0.9 V , bo vsota napetosti $0.9 \text{ V} + 0.2 \text{ V} = 1.1 \text{ V}$ že v prepovedanem območju.



Slika 2.6: Model povezave med dvema vezjema, ki upošteva inducirani šum

Slika 2.6 prikazuje bolj realen model povezave med dvema integriranimi vezjema. Odpornost na šum naredimo tako, da določimo ožje območje veljavnih potencialov na izhodni strani in širše na vhodni strani, kot prikazuje slika 2.7. Razlika v širini območja se imenuje šumna meja in zagotavlja določeno odpornost na induciran šum pri komunikaciji.



Slika 2.7: Statični red z upoštevanjem šumne meje

Veljavno območje potencialov na oddajni in sprejemni strani je sedaj podano z enačbami:

	<i>oddajnik</i>	<i>sprejemnik</i>
logična '0':	$0V \leq V_L \leq V_{OL}$	$0V \leq V_L \leq V_{IL}$
logična '1':	$V_{OH} \leq V_H \leq V_{dd}$	$V_{IH} \leq V_H \leq V_{dd}$

Vrednosti potencialov določajo standardi, npr. TTL, CMOS, LVCMOS. Izjava proizvajalca digitalnih integriranih vezij o skladnosti s standardom zagotavlja, da bomo brez težav povezali signale različnih vezij med seboj.

Inducirani šum lahko povzroči, da bo vrednost signala izven meja napajalnih napetosti: višja od Vdd ali nižja od GND (negativna napetost). Takšen signal se na sprejemniku sicer pravilno interpretira, lahko pa povzroči uničenje vezja, če preseže določene meje.

Primer vrednosti potencialov za vezja CMOS z napajalno napetostjo 5 V podaja tabela 2.2, za vezja v 3.3 V tehnologiji LVCMOS pa tabela 2.3.

oznaka	pomen	napetost[V]
V_{IH}	vhodni visok nivo	3
V_{IL}	vhodni nizek nivo	1
V_{OH}	izhodni visok nivo	3.1
V_{OL}	izhodni nizek nivo	0.2

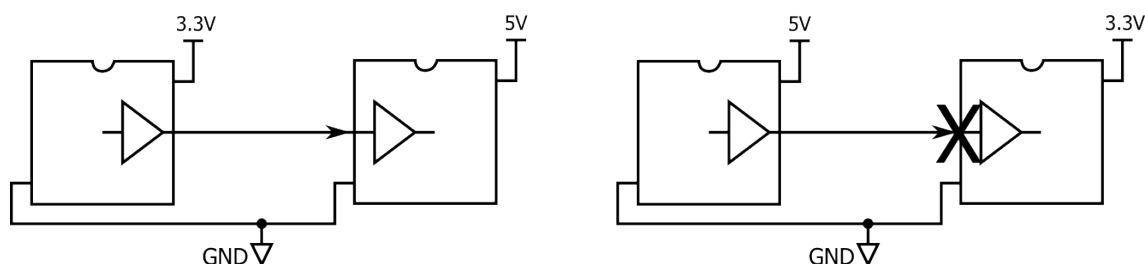
Tabela 2.2: Statični parametri vezij CMOS pri Vdd = 5V

oznaka	pomen	<i>min</i>	<i>max</i>
V_{dd}	napajalna napetost	3.0V	3.6V
V_{IH}	vhodni visok nivo	2V	$V_{dd} + 0.3V$
V_{IL}	vhodni nizek nivo	-0.3V	+0.8V
V_{OH}	izhodni visok nivo	$V_{dd} - 0.2V$	
V_{OL}	izhodni nizek nivo		+0.2V

Tabela 2.3: Statični parametri LVCMOS pri normalnem razponu V_{dd} (JEDEC)

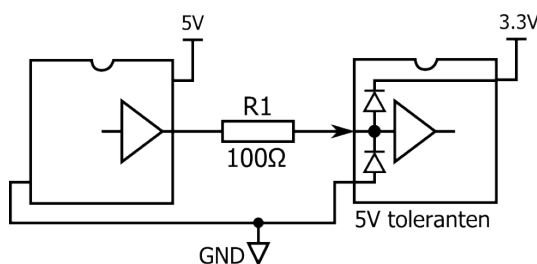
2.1.2 Povezovanje različnih standardov

Včasih potrebujemo povezavo med integriranimi vezji, ki uporabljajo različne standarde. Pogledjmo si primer povezave 5V CMOS in 3.3V LVCMOS vezja. Izhod 3.3V vezja lahko brez težav krmili vhod 5V vezja, kot prikazuje leva stran slike 2.8. Visoko stanje LVCMOS je vsaj 3.1V, kar je dovolj velik potencial, da se interpretira kot logična '1' v drugem vezju. Nizko stanje na izhodu je največ 0.2V, kar ponovno zadošča za 5V CMOS vhode. Zavedati se moramo le, da smo s takšno povezavo znižali šumno mejo.



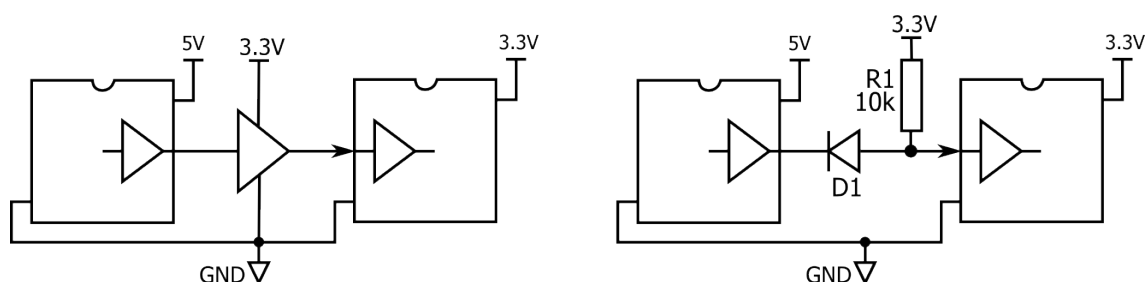
Slika 2.8: Povezovanje vezij 5V CMOS in 3.3V LVCMOS

Pri obratni povezavi iz 5V izhoda na vhod 3.3V logike moramo biti precej bolj previdni in jo brez pregleda specifikacij elementov ne smemo narediti. Visoko stanje 5V logike je namreč precej nad napajalnim nivojem LVCMOS, zato bo po signalni liniji stekel velik enosmerni tok čez vhod vezja proti 3.3V napajalni liniji. To je posledica delovanja zaščitnih diod na vseh vhodih vezja, ki pri previsoki napetosti kratko sklenejo vhod proti napajalni povezavi. Tok skozi vhod omejimo z zaporedno vezanim zaščitnim upornikom:



Slika 2.9: Povezava 5V CMOS na 3.3V LVCMOS preko zaščitnega upornika

Tipična upornost zaščitnega upornika je nekaj 100Ω in je kompromis med visoko vrednostjo, ki povzroči počasnejši prenos signala in nizko, pri kateri teče višji tok. Poleg omejitve toka je potrebno upoštevati tudi omejitve napetosti, saj sodobna vezja uniči že statična napetost na vhodu. Trend v tehnologiji integriranih vezij gre namreč proti zmanjševanju dimenzij in strukture v vezju so tako majhne, da hitro pride do preboja, ki jih trajno poškoduje. Sodobna integrirana vezja lahko poškoduje že napetost višja od 4 V. Če želimo takšno vezje priklopiti na izhod CMOS, ne zadošča le zaščitni upornik. Najboljša rešitev je z uporabo namenskega ojačevalnika za pretvorbo potencialnih nivojev, za manj zahtevne primere pa naredimo vezje za omejitev napetosti iz pasivnih elementov:



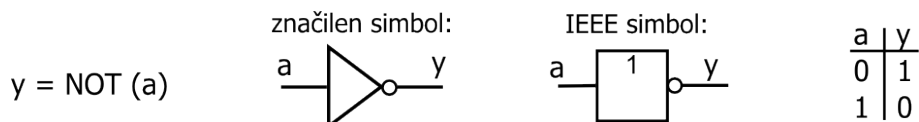
Slika 2.10: Pretvorba logičnih nivojev za občutljiva vezja LVC MOS

Dioda D1 prevaja, ko je na izhodu oddajnega vezja nizko stanje. V tem primeru je na diodi napetost okoli 0.7V, ki jo zazna vezje kot logično '0' na vhodu. Kadar je na izhodu napetost 5V, je dioda zaprta in na vhod drugega vezja pride preko upora R1 napetost 3.3V. Vrednost upora R1 naj bo nekaj $k\Omega$.



2.2 Logična vrata

Logična vrata so osnovni gradniki digitalnih vezij. Uporabljajo se za izvedbo logičnih funkcij nad binarnimi vrednostmi, ki jih prenašajo digitalni signali. Logična vrata izhajajo iz matematične predstavitve osnovnih operacij nad binarnimi števili, ki jih obravnava Boolova algebra. Najbolj preprosta operacija je negacija binarne vrednosti: Boolov izraz NOT(a) ima vrednost 1, kadar je $a=0$, pri $a=1$ pa ima vrednost 0. Načrtovanje kombinacijskega digitalnega vezja začnemo z opisom problema, ki ga prevedemo v Boolove izraze. Nato narišemo shemo vezja z uporabo grafičnih simbolov logičnih funkcij.



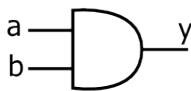
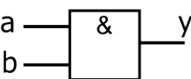
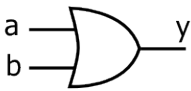
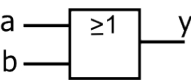
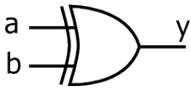
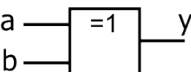
Slika 2.11: Boolova negacija, simbol negatorja in pravilnostna tabela.

Slika 2.11 prikazuje dve obliki grafičnega simbola negatorja: značilen trikotni simbol in standardni simbol v obliki pravokotnika (standard IEEE std. 91-1984). Oba simbola imata majhen

krožec na izhodnem priključku, ki označuje negiranje oz. inverzijo logične vrednosti. Pri risanju logičnih shem z računalniškimi orodji bomo večkrat naleteli na značilno obliko simbola, zato bomo to obliko uporabljali tudi v našem gradivu. Delovanje negatorja predstavimo s *pravilnostno tabelo* v kateri so določene vrednosti izhodov pri različnih stanjih vhoda.

Poglejmo še nekaj Boolovih operacij nad dvema binarnima signaloma: IN, ALI in ekskluzivni ALI:

- **a AND b** ima vrednost 1, kadar je prvi IN drugi operand 1, sicer ima vrednost 0.
- **a OR b** ima vrednost 1, kadar je vsaj en operand a ALI b enak 1. Vrednost 0 pa ima ko sta oba operanda 0.
- **a XOR b** ima vrednost 1, kadar sta operanda različna, sicer ima vrednost 0.

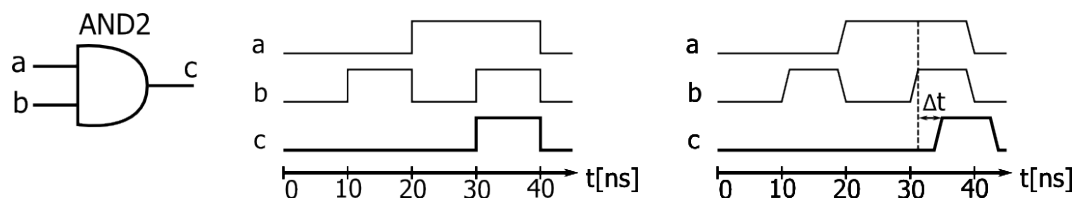
operacija	grafični simbol	pravilnostna tabela															
$y = a \text{ AND } b$	značilen: 	<table><tr><th>a</th><th>b</th><th>y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	a	b	y	0	0	0	0	1	0	1	0	0	1	1	1
	a	b	y														
0	0	0															
0	1	0															
1	0	0															
1	1	1															
	IEEE: 																
$y = a \text{ OR } b$	značilen: 	<table><tr><th>a</th><th>b</th><th>y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	a	b	y	0	0	0	0	1	1	1	0	1	1	1	1
	a	b	y														
0	0	0															
0	1	1															
1	0	1															
1	1	1															
	IEEE: 																
$y = a \text{ XOR } b$	značilen: 	<table><tr><th>a</th><th>b</th><th>y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	a	b	y	0	0	0	0	1	1	1	0	1	1	1	0
	a	b	y														
0	0	0															
0	1	1															
1	0	1															
1	1	0															
	IEEE: 																

Slika 2.12: Boolove operacije, simboli logičnih vrat in pravilnostne tabele.

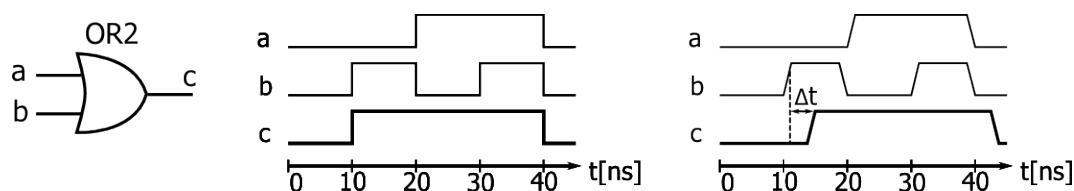
Slika 2.12 prikazuje grafične simbole teh operacij v značilni in standardni obliki ter njihove pravilnostne tabele. Ker imamo dva vhodna signala, so v pravilnostni tabeli štiri vrstice s katerimi predstavimo vsa stanja na vhodih: 0 0, 0 1, 1 0 in 1 1.

Delovanje logičnih vrat lahko opazujemo tudi na časovnem diagramu, kjer s časom spreminjamo stanje vhodov in opazujemo izhod. Logični simulator običajno ne upošteva zakasnitev vrat, zato se izhod spremeni takoj ob spremembi vhodov. V realnih logičnih vratih se izhod ne spremeni takoj ampak z neko zakasnitvijo, ki je posledica preklopnih časov elektronskih elementov iz katerih so logična vrata narejena.

Sliki 2.13 in 2.14 prikazujeta časovne diagrame logičnih vrat v obliki simulacije in v obliki realnega diagrama z zakasnitvijo.



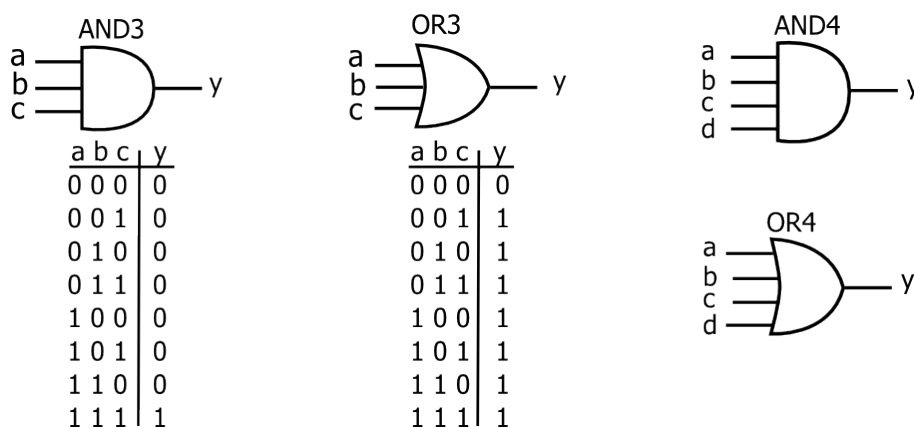
Slika 2.13: Idealni in realni časovni diagram logičnih vrat AND2.



Slika 2.14: Idealni in realni časovni diagram logičnih vrat OR2.

Na realnih časovnih diagramih smo označili zakasnitev z Δt . Zakasnitve posledično omejujejo hitrost spreminjanja signalov na vseh vhodi in s tem hitrost delovanja vezja. Kadar so majhne v primerjavi s pričakovanimi časovnimi intervali spreminjanja signalov, jih lahko ignoriramo. Za analizo vezja pogosto zadostuje idealni časovni diagram, v katerem zakasnitve niso prikazane.

Logična vrata imajo lahko tudi več kot 2 vhodna signala, kot prikazuje slika 2.15. Izhod večvhodnih vrat AND je 1 v primeru, ko so vsi vhodi enaki 1, sicer je izhod enak 0. Izhod vrat OR pa je 1 kadar je vsaj eden izmed vhodov enak 1, 0 pa je le primeru, ko so vsi vhodi enaki 0.



Slika 2.15: Logična vrata AND in OR z več vhodi.

S predstavljenimi operacijami lahko zapišemo enačbe v Boolovi aritmetiki za reševanje poljubne logične naloge. Izkaže se, da lahko vse logične funkcije izrazimo s kombinacijo osnovnih operacij: AND, OR in NOT. To lastnost izkoriščajo nekatera programirljiva integrirana

vezja, ki jim v postopku programiranja določimo povezave med osnovnimi operacijami in tako omogočimo, da izvajajo poljubno digitalno funkcijo. Operacija XOR pa je osnovni element aritmetičnih gradnikov za seštevanje, odštevanje in primerjavo večbitnih logičnih vrednosti.

2.3 Načrtovanje logičnih vezij



Postopek načrtovanja preprostih digitalnih vezij, ki izvajajo logične operacije za rešitev podane naloge bomo predstavili na nekaj primerih. Vsak primer se začne z opisom naloge, ki ga pretvorimo v Boolove izraze in nato v shemo logičnega vezja.

2.3.1 Avtomobilski alarm

Avtomobilski alarm naj se sproži, kadar je nastavljen in kadar je izpolnjen vsaj eden izmed pogojev: ali je aktiviran senzor gibanja ali pa so odprta vrata. Signal *alarm* naj predstavlja stanje alarma, signal *vklop* določa ali je alarm nastavljen, signal *vrata* naj predstavlja stanje vrat (stanje 1 pomeni odprta) in signal *gib* stanje senzorja. Delovanje alarma opišemo z logično enačbo:

$$alarm = vklop \text{ AND } (vrata \text{ OR } gib)$$

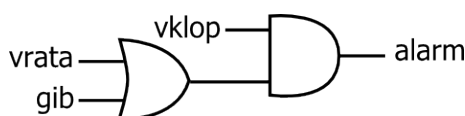
Sedaj lahko računamo vrednosti signala *alarm* pri različnih vhodnih vrednostih:

- pri $vklop = 1, vrata = 0, gib = 0$ dobimo $alarm = 1 \text{ AND } (0 \text{ OR } 0) = 1 \text{ AND } 0 = 0$
- pri $vklop = 1, vrata = 1, gib = 0$ dobimo $alarm = 1 \text{ AND } (1 \text{ OR } 0) = 1 \text{ AND } 1 = 1$
- pri $vklop = 0, vrata = 1, gib = 0$ dobimo $alarm = 0 \text{ AND } (1 \text{ OR } 0) = 0 \text{ AND } 1 = 0$

Na podlagi logične enačbe lahko dokažemo nekatere trditve. Npr. trditev, da se alarm ne bo sprožil, če je vklop na 0. Logični izraz **AND** bo imel avtomatsko vrednost 0, če je na eni strani vrednost 0:

$$alarm = 0 \text{ AND } (vrata \text{ OR } gib) = 0$$

Logično enačbo pretvorimo v shemo vezja v par korakov. Najprej narišemo signal *alarm*, ki je izhod iz logičnih vrat **AND**. Vhod teh vrat je enkrat signal *vklop*, drugič pa izhod vrat **OR**, ki imajo na vhodu signala *vrata* in *gib*, kot prikazuje slika 2.16.



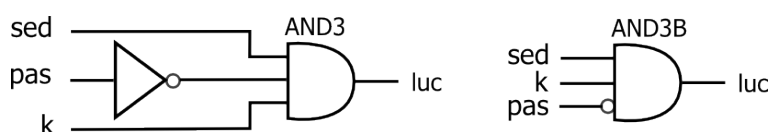
Slika 2.16: Shema vezja za avtomobilski alarm.

2.3.2 Indikator varnostnega pasu

Naredimo vezje, ki prižge opozorilni indikator, če nismo pripeti z varnostnim pasom. Signal s senzorja varnostnega pasu (*pas*) je v stanju 1, ko je pas pripet in v stanju 0, kadar je odpet. Senzor v sedežu postavi signal *sed* v stanje 1, kadar nekdo sedi v avtu. Poleg tega imamo informacijo ali je ključ (*k*) v položaju, ko je avto v pogonu (stanje 1). Indikatorska luč naj se prižge, če voznik sedi v avtu IN je pas odpet IN je avto v pogonu, kar zapišemo z enačbo:

$$luc = sed \text{ AND NOT}(pas) \text{ AND } k$$

Logično vezje naredimo z enimi tri-vhodnimi logičnimi vrati AND in negatorjem, ali pa s posebnim simbolom vrat AND, kjer negiran vhod označimo s krogcem, kot prikazuje slika 2.17.

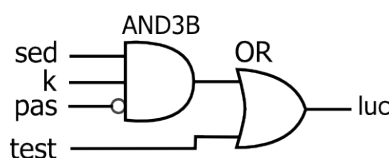


Slika 2.17: Dve obliki sheme vezja za opozorilno luč varnostnega pasu.

Opozorilni indikatorji v avtomobilu se prižgejo za kratek čas tudi kadar obrnemo ključ. S tem preverimo ali vse opozorilne lučke res delujejo. Vzemimo, da imamo na voljo testni signal *test*, ki se ko obrnemo ključ za nekaj sekund postavi na 1, potem pa gre nazaj na 0. Indikatorska luč naj se prižge kadar je testni signal na 1 ALI pa v primeru, če voznik ni pripet:

$$luc = test \text{ OR } sed \text{ AND NOT}(pas) \text{ AND } k$$

Shemo digitalnega vezja indikatorja s testnim signalom prikazuje slika 2.18.



Slika 2.18: Končna shema vezja za opozorilno luč varnostnega pasu.

2.3.3 Ugibaj kombinacijo

Naredimo enostavno igro, pri kateri en igralec nastavi kombinacijo dveh signalov: a in b, drugi pa poskuša nastavljeno kombinacijo uganiti s postavljanjem svojih dveh signalov: c in d. Ko sta nastavljeni kombinaciji enaki, naj se vklopi indikatorska dioda.

Naloga se lotimo tako, da najprej rešimo enostavnejši problem primerjave dveh signalov. Uporabili bomo lastnost ekskluzivne ali operacije, ki postavi izhod na 1, kadar sta vhodna signala različna. Ker želimo postaviti izhod na 1, kadar sta vhodna signala enaka, moramo le še negirati izhod.

Zapišimo Boolove izraze za primerjavo signalov a in c ter b in d:

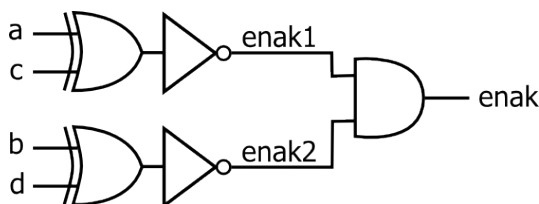
$$enak1 = \text{NOT}(a \text{ XOR } c)$$

$$enak2 = \text{NOT}(b \text{ XOR } d)$$

Sedaj ni težko rešiti prvotne naloge: kombinaciji signalov sta enaki, kadar je postavljen na 1 rezultat prve primerjave IN druge primerjave:

$$enak = enak1 \text{ AND } enak2$$

Slika 2.19 prikazuje ustrezno logično vezje, ki se v terminologiji digitalnih vezij imenuje primerjalnik. Na podoben način bi naredili primerjavo enakosti večbitnih vrednosti.



Slika 2.19: Vezje za primerjavo dveh 2-bitnih vhodnih vrednosti.



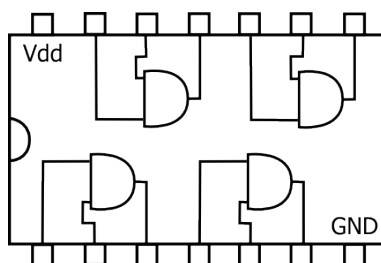
2.4 Izdelava vezij z logičnimi vrati

Logična vrata so narejena z elektronskimi stikali. Prva digitalna vezja so bila narejena iz elektromehanskih stikal - relejev, ki so jih kasneje zamenjale elektronske cevi (elektronke). To so bili veliki in okorni gradniki, ki so bili pogosto v okvari. Leta 1945 so med iskanjem napake v digitalnem računalniku Mark II našli hrošča med kontakti releja. Od takrat računalniške napake imenujejo hrošč (angl. bug) in postopek odkrivanja ter odpravljanja napak "razhroščevanje" (debugging). Releje in elektronke so nadomestili precej manjši in bolj zanesljivi transistorji. Največji skok v razvoju digitalnih sistemov pa so omogočila *integrirana vezja*, v katerih je na majhni rezini silicija narejeno celotno vezje.

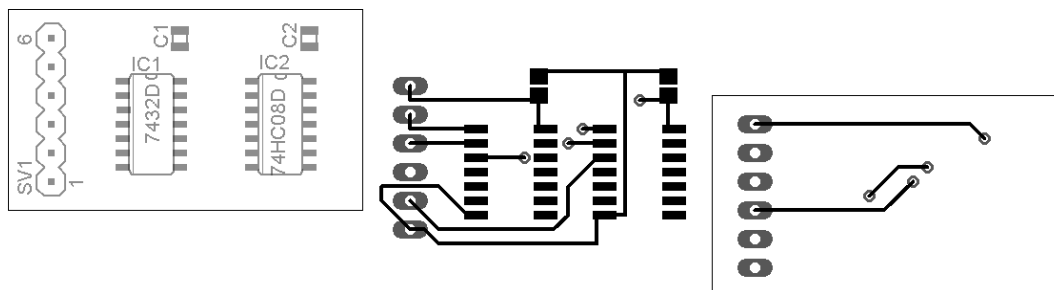
2.4.1 Integrirana vezja

Proizvajalci integriranih vezij so pripravili celo vrsto digitalnih integriranih vezij za potrebe razvijalcev digitalnih sistemov. Med njimi najdemo osnovna logična vrata, pomnilnike in sestavljene gradnike za najbolj pogosto uporabljene naloge. Integrirana vezja, ki so narejena v določenem proizvodnem procesu iz enakih elektronskih elementov imenujemo družina integriranih vezij. Najbolj znana družina digitalnih integriranih vezij ima oznako 74nnn, kjer številčna oznaka nnn določa funkcijo posameznega vezja.

Slika 2.20 prikazuje integrirano vezje z oznako 7408 v katerem so štiri dvovhodna logična vrata AND, na sliki 2.21 pa je načrt tiskanega vezja za avtomobilski alarm, ki vsebuje dve integrirani vezji z logičnimi vrati.



Slika 2.20: Integrirano vezje 7408 s štirimi logičnimi vrati AND.



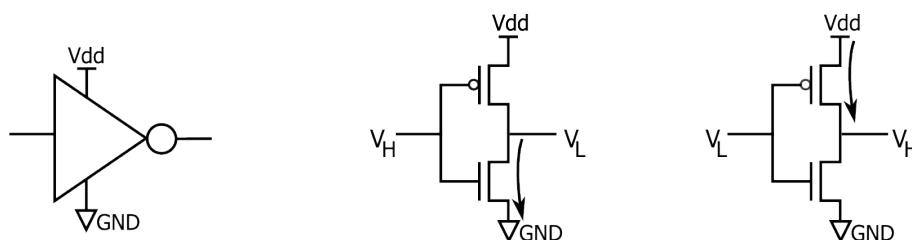
Slika 2.21: Načrt tiskanega vezja za avtomobilski alarm: sitotisk, zgornja in spodnja plast.

2.4.2 Elektronska stikala

Digitalna integrirana vezja vsebujejo elektronska stikala, ki so narejena s polprevodniškimi *transistorji*. Transistor deluje kot stikalo, pri katerem električno prevodnost med dvema kontaktoma spreminjamo z električnim potencialom na kontrolnem vhodu.

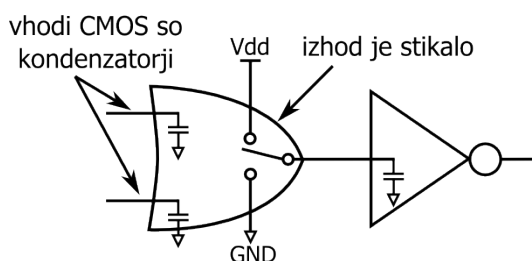
Digitalna vezja z oznako CMOS so zgrajena iz komplementarnih nMOS in pMOS transistorjev. Slika 2.22 prikazuje zgradbo logičnega negatorja v CMOS izvedbi. Kadar je na vhodu negatorja visok potencial, bo prevajal spodnji transistor (nMOS) in povezal izhod na maso oz. potencial V_L . Če je na vhodu nizek potencial pa prevaja zgornji transistor (pMOS) in takrat je na izhodu napajalna napetost oz. potencial V_H .

Zgornji in spodnji transistor sta odprta izmenično, zato pri konstantnem vhodu tudi skozi napajalni sponki CMOS gradnika praktično tok ne teče. Digitalni gradniki v izvedbi CMOS imajo izredno majhno porabo in so idealni za izdelavo kompleksnih vezij. Tehnologija integriranih vezij omogoča izdelavo vedno manjših transistorjev oz. vedno večjega števila transistorjev na enaki površini. Digitalna integrirana vezja vsebujejo danes na milijone polprevodniških transistorjev, ki delujejo kot stikala.



Slika 2.22: Logični negator v CMOS izvedbi.

Pri obravnavi digitalnih vezij bomo uporabljali poenostavljene modele. Kontrolni vhodi CMOS transistorjev se obnašajo kot kondenzatorji, skozi katere enosmerni tok ne teče. Za enosmerne napetosti in tokove predstavlja takšen vhod odprte sponke, tako da enosmerni tok preko vhodnega signala ne teče. Izhod predstavimo zelo poenostavljeno s stikalom, ki preklaplja izhodni signal med napajalnima nivojema V_{dd} in GND . V resnici izhod ni idealno stikalo, ampak ima neko upornost, saj je tok skozi izhodni preklopni element omejen (običajno $10 - 20mA$). Slika 2.23 predstavlja električni model digitalnih vhodnih in izhodnih sponk vezja v tehnologiji CMOS.



Slika 2.23: Poenostavljen model vhoda in izhoda v tehnologiji CMOS

Naloge

1. Na vhod logičnega gradnika s statičnimi parametri: $V_{dd} = 3.3V$, $V_{IH} = 2V$, $V_{IL} = 0.8V$ pripeljemo signal z napetostjo $U = 2.5V$ proti masi. Kako bo gradnik ta signal intepretiral:
 - (a) kot visoko stanje (logična 1)
 - (b) kot nizko stanje (logična 0)
 - (c) kot nedovoljeno stanje
 - (d) kot kratek stik
2. Preglej obe logični shemi in ugotovi v katerem položaju stikala bo LED prižgana!

