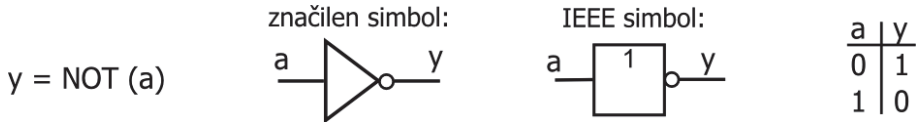
3
Logična vrata

Logična vrata so osnovni gradniki digitalnih vezij. Uporabljajo se za izvedbo logičnih funkcij nad binarnimi vrednostmi, ki jih prenašajo digitalni signali.

3.1 Osnovna logična vrata

Logična vrata izhajajo iz matematične predstavitve osnovnih operacij nad binarnimi števili, ki jih obravnava Boolova algebra. Najpreprostejša 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.

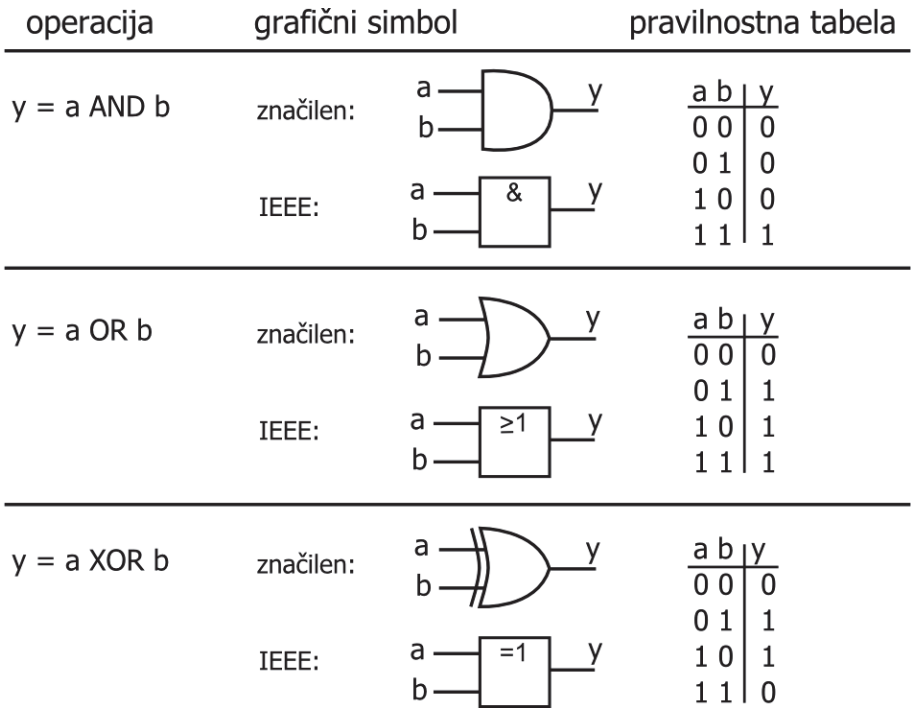


Slika 3.1: Boolova negacija, simbol negatorja in pravilnostna tabela.

V električnih shemah zasledimo 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 vseh možnih stanjih vhoda.

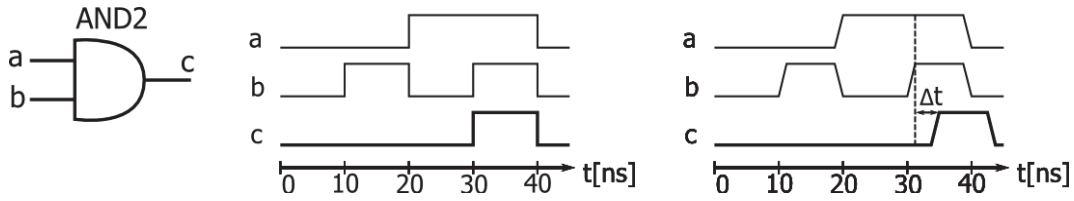
Poglejmo še nekaj Boolovih operacij nad dvema binarnima signaloma: *logična in* (a **AND** b) je 1, kadar sta prvi in drugi operand 1, sicer ima vrednost 0. Operacija *logični ali* (a **OR** b) je 1, kadar je vsaj en operand enak 1. Vrednost 0 ima le, ko sta oba operanda 0. Operacija *ekskluzivni ali* (a **XOR** b) je 1, kadar sta operanda različna, sicer pa ima vrednost 0.

Za vsako operacijo obstaja grafični simbol v značilni in standardni obliki, ki predstavlja ustrezna logična vrata. Delovanje teh operacij opisuje pravilnostna tabela s štirimi vrsticami, ker imamo štiri možne kombinacije vhodnih signalov: 0 0, 0 1, 1 0 in 1 1.

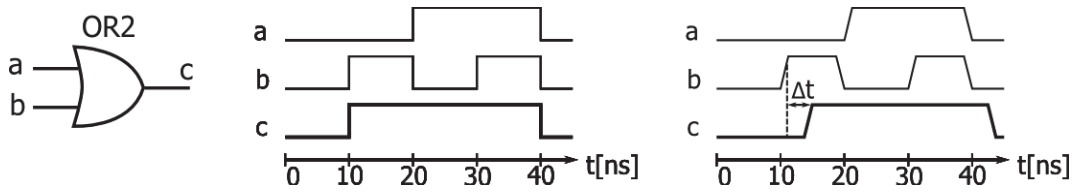


Slika 3.2: Boolove operacije, simboli logičnih vrat in pravilnostne tabele.

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 določeno zakasnitvijo, ki je posledica preklopnih časov elektronskih elementov, iz katerih so logična vrata narejena.



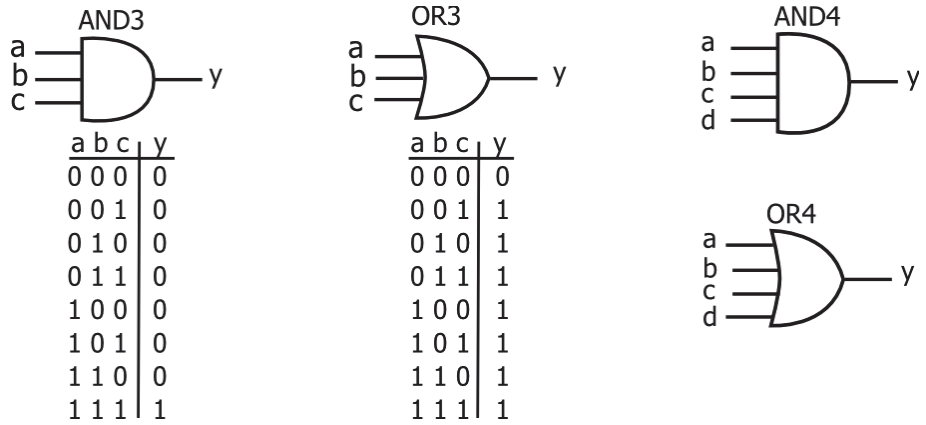
Slika 3.3: Idealni in realni časovni diagram logičnih vrat AND2.



Slika 3.4: Idealni in realni časovni diagram logičnih vrat OR2.

Na realnih časovnih diagramih smo označili zakasnitev z Δt. Zakasnitve posledično omejujejo hitrost spreminjanja signalov na vhodih 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 dva vhodna signala. 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 3.5: 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.

3.2 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.

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:

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

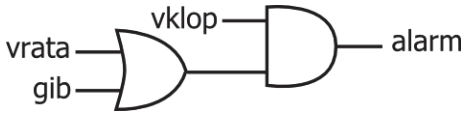
Sedaj lahko računamo vrednosti signala $alarm$ pri različnih vhodnih vrednostih:

- 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 nekaj korakih. Najprej narišemo signal $alarm$, ki je izhod z logičnih vrat **AND**. Vhod teh vrat je enkrat signal $vklop$, drugič pa izhod vrat **OR**, ki imajo na vhodu signala $vrata$ in gib.



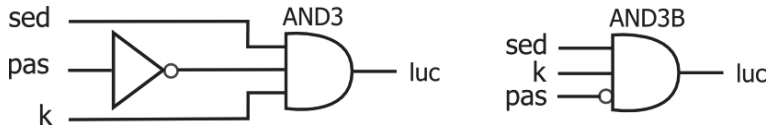
Slika 3.6: Shema vezja za avtomobilski alarm.

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 se prižge, če voznik sedi v avtu *in* je pas odpet *in* avto v pogonu, kar zapišemo z enačbo:

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

Logično vezje naredimo s trivhodnimi logičnimi vrati AND in negatorjem ali pa s posebnim simbolom vrat AND, kjer negiran vhod označimo s krožcem.

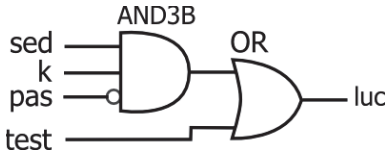


Slika 3.7: Dve obliki sheme vezja za opozorilno luč varnostnega pasu.

Opozorilni avtomobilski indikatorji se prižgejo za kratek čas tudi, kadar obrnemo ključ. S tem preverimo, ali 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 se prižge, kadar je testni signal na 1 *ali* 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 3.8.



Slika 3.8: Končna sheme vezja za opozorilno luč varnostnega pasu.

Ugibaj kombinacijo

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

Naloge 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:

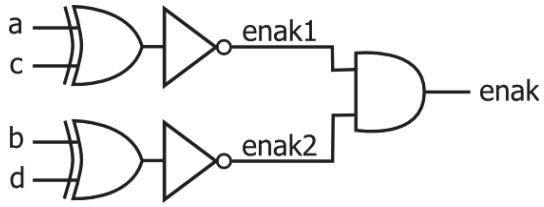
$$enak1 = \text{NOT}(a \text{ XOR } c)$$

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

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



Slika 3.9: Vezje za primerjavo dveh 2-bitnih vhodnih vrednosti.

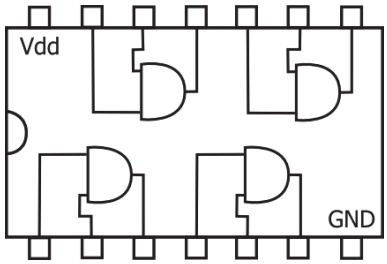
3.3 Izdelava vezij z logičnimi vrati

Logična vrata so narejena z elektronskimi stikali. Prva digitalna vezja so bila narejena iz elektromehanskih stikal – relejev, ki so jih pozneje 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 zanesljivejši tranzistorji. Največji skok v razvoju digitalnih sistemov pa so omogočila *integrirana vezja*, v katerih je na majhni rezini silicija narejeno celotno vezje.

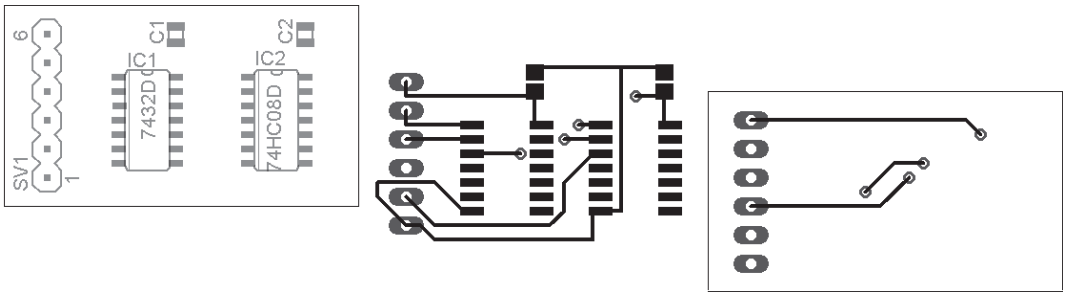
Integrirana vezja

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 najpogosteje uporabljene naloge. Digitalna integrirana vezja, ki so narejena v določenem proizvodnem procesu, z med seboj združljivimi statičnimi parametri, imenujemo *družina* logičnih vezij. Najbolj znana je družina logičnih vezij 7400, ki vsebuje predstavnike vseh pomembnejših logičnih gradnikov.

Slika 3.10 prikazuje integrirano vezje z oznako 7408, v katerem so štiri dvovhodna logična vrata AND, na sliki 3.11 pa je načrt tiskanega vezja za avtomobilski alarm, ki vsebuje dve integrirani vezji z logičnimi vrati.



Slika 3.10: Integrirano vezje 7408 s štirimi logičnimi vrati AND.



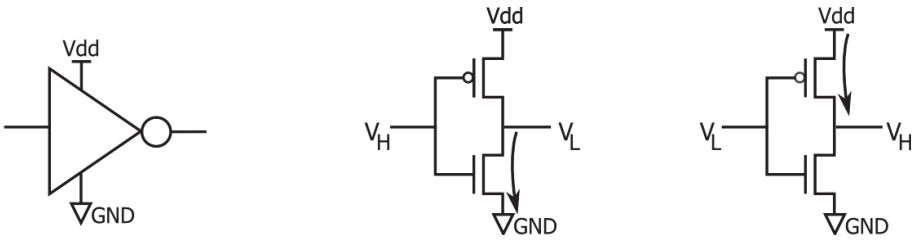
Slika 3.11: Načrt tiskanega vezja za avtomobilski alarm (sitotisk, zgornja in spodnja plast).

Elektronska stikala

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

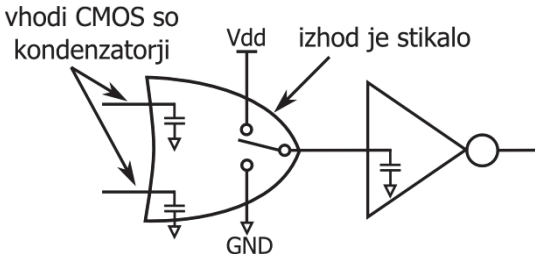
Digitalna vezja z oznako CMOS so zgrajena iz komplementarnih tranzistorjev vrste nMOS in pMOS. Slika 3.12 prikazuje zgradbo logičnega negatorja v izvedbi CMOS. Kadar je na vhodu negatorja visok potencial, bo prevajal spodnji tranzistor (nMOS) in povezal izhod na maso oz. potencial V_L. Če je na vhodu nizek potencial pa prevaja zgornji (pMOS) in takrat je na izhodu napajalna napetost oz. potencial V_H.

Zgornji in spodnji tranzistor sta odprta izmenično, zato pri konstantnem vhodu tudi skozi napajalni sponki gradnika CMOS 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 tranzistorjev oz. vedno večjega števila le-teh na enaki površini. Digitalna integrirana vezja vsebujejo danes na milijone polprevodniških tranzistorjev, ki delujejo kot stikala.



Slika 3.12: Logični negator v CMOS izvedbi.

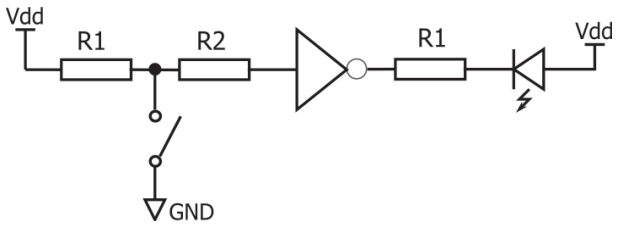
Pri obravnavi digitalnih vezij bomo uporabljali poenostavljene modele. Kontrolni vhodi tranzistorjev CMOS se obnašajo kot kondenzatorji, skozi katere enosmerni tok ne teče. Za enosmerne napetosti in tokove predstavlja takšen vhod odprte sponke, tako da enosmerni tok prek vhodnega signala ne teče. Izhod 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 3.13 predstavlja električni model digitalnih vhodnih in izhodnih sponk vezja v tehnologiji CMOS.



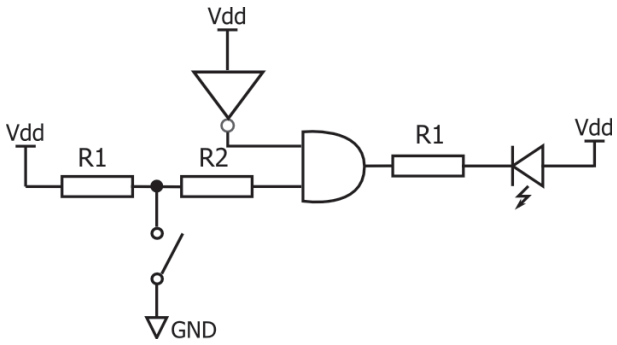
Slika 3.13: Poenostavljen model vhoda in izhoda v tehnologiji CMOS

Naloge

1. Preglej obe logični shemi in ugotovi, v katerem položaju stikala bo LED prižgana.



2. Poglej vezavo na shemi in ugotovi, kdaj bo LED prižgana.



3. Utemelji, zakaj ne smemo vezati skupaj izhodov dveh logičnih vrat. Pomagaj si s poenostavljenim modelom logičnega gradnika v tehnologiji CMOS.