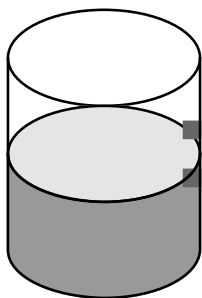


# OSNOVNA LOGIČKA KOLA

## Zadatak 1

U jednoj banci čuva se vredan dijamant. Dijamant se nalazi na posebnom postolju osetljivom na potrese. Postolje se sastoji od dva cilindra delimično napunjena tečnošću, od kojih svaki ima dva senzora tečnosti postavljene neposredno ispod i iznad površine tečnosti. Kada se senzor nalazi u tečnosti, logička vrednost signala senzora je "1", a kada se nalazi van tečnosti, logička vrednost signala senzora je "0".



Slika 1.1: Izgled jednog cilindra

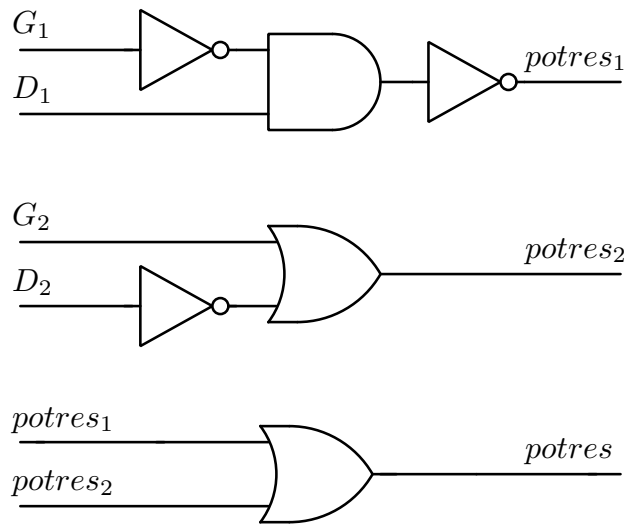
Potrebno je napraviti signal *potres* koji na osnovu signala senzora prijavljuje potres i to ukoliko bilo koji od cilindara prijavljuje potres. U slučaju da jedan cilindar prijavljuje potres a drugi ne, signalizirati kvar postavljanjem signala *kvar* na logičku vrednost "1". U slučaju registrovanja potresa, treba uključiti zvono alarma postavljanjem signala *alarm* na vrednost "1". Kada se vrši čišćenje dijamanta, vrši se deaktivacija zvona alarma. Tada signal *iskljuci\_zvono* ima vrednost "1" i ne treba uključiti zvono čak i ako se detektuju potresi.

Banka želi da u budućnosti možda promeni senzore, ali da ne menja i ostatak elektronike. Kako nisu sigurni da li će budući senzori imati vrednost "1" u vazduhu ili vodi, potrebno je napraviti sistem koji će raditi u oba slučaja.

## **REŠENJE:**

Pokušajmo da napravimo signal  $potres_1$  koji će imati logičku vrednost "1" ukoliko je došlo do potresa prvog cilindra, odnosno "0" ukoliko nije došlo do potresa. Ukoliko su signali gornjeg i donjeg senzora prvog cilindra  $G_1$  i  $D_1$ , tada je jedina kombinacija ovih signala u slučaju kada nema potresa  $G_1 = 0$ ,  $D_1 = 1$ . Dakle, nije bilo potresa samo ako je  $G_1 = 0$  i  $D_1 = 1$ . Šema koja prikazuje ovu zavisnost je data slikom 1.2.

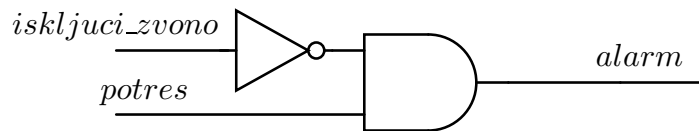
Signal  $potres_2$  možemo dobiti na identičan način. Možemo rasuđivati i malo drugačije i reći da je potres registrovan ukoliko je gornji senzor registrovao vodu ( $G_2 = 1$ ) ili ukoliko je donji senzor registrovao vazduh ( $D_2 = 0$ ).



Slika 1.2: Realizacija signala *potres*

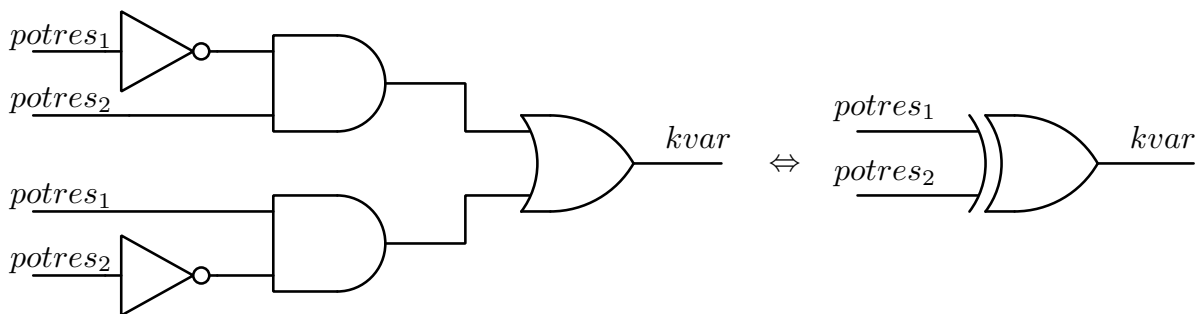
Potres treba registrovati ukoliko je bilo koji od senzora registrovao potres.

Signal *alarm* treba da ima vrednost "1" samo ukoliko je došlo do potresa i ukoliko nije isključeno zvono (slika 1.3).



Slika 1.3: Realizacija signala *alarm*

Signal *kvar* treba generisati ukoliko su *potres1* i *potres2* različiti. Signali *potres1* i *potres2* su različiti u dva slučaja: ukoliko je detektovan *potres1* i *potres2* nije ili ukoliko nije detektovan *potres1* i *potres2* jeste (slika 1.4).



Slika 1.4: Realizacija signala *kvar*

Iako je gornje rešenje intuitivno blisko, pokušajmo da ga sagledamo na jedan drugačiji način. Primetimo da I kolo kao izlaz daje "0" uvek osim za jednu kombinaciju ulaza – kada su svi ulazni signali "1". Takođe ILI kolo kao izlaz daje "1" uvek osim za jednu kombinaciju ulaza – kada su svi ulazni signali "0". Ovo možemo shvatiti kao da i I i ILI kola izdvajaju jednu kombinaciju ulaza od svih ostalih mogućih kombinacija ulaza. Ako se vratimo na generisanje signala *potres1*, mi smo želeli da od svih mogućih ulaznih kombinacija signala  $G_1$  i  $D_1$ , samo jednu od njih izdvojimo od ostalih ( $G_1 = 0, D_1 = 1$ , jer nam je ta kombinacija predstavljala stanje u kome nema potresa). Ukoliko bismo želeli da iskoristimo I kolo da izdvojimo ovu kombinaciju ulaza od svih ostalih kombinacija ulaza, morali bi ovu kombinaciju transformišemo u onu kombinaciju koju I kolo izdvaja – kombinaciju 11. Invertovanjem signala  $G_1$  upravo smo to i dobili. Sada se

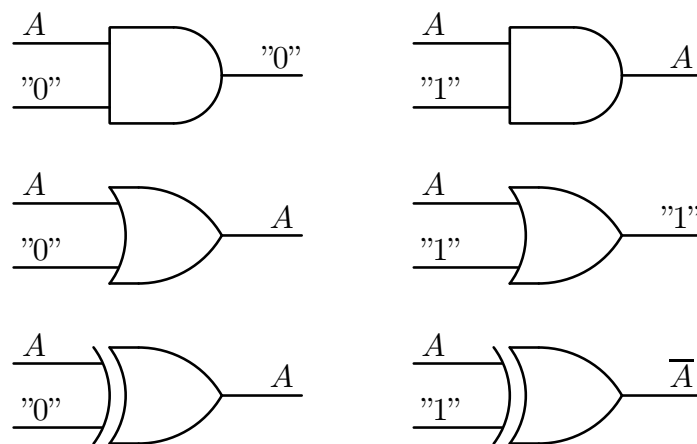
od svih kombinacija signala  $G_1$  i  $D_1$  samo jedna transformiše u kombinaciju gde su svi ulazni signali I kola "1", baš kombinacija  $G_1 = 0, D_1 = 1$ . Dakle, na izlazu I kola smo dobili logičku vrednost "0" za sve kombinacije signala  $G_1, D_1$  osim za kombinaciju  $G_1 = 0, D_1 = 1$  kada je izlaz I kola "1". Nama međutim treba da je vrednost signala  $potres_1 = 1$  za sve kombinacije signala  $G_1, D_1$  osim za kombinaciju  $G_1 = 0, D_1 = 1$  kada treba da je vrednost signala  $potres_1 = 0$ . Invertovanjem izlaznog signala I kola to smo i dobili.

U realizaciji signala  $potres_2$  koristili smo ILI kolo, koje od svih mogućih kombinacija ulaza izdvaja kombinaciju kada su svi ulazi "0". Invertovanjem signala  $D_1$  dobili smo da od svih kombinacija signala  $G_1$  i  $D_1$  samo jedna odgovara kombinaciji ulaza ILI kola kada su svi ulazi "0" – upravo kombinacija  $G_1 = 0, D_1 = 1$ . Dakle, na izlazu ILI kola smo dobili logičku vrednost "1" za sve kombinacije signala  $G_1, D_1$  osim za kombinaciju  $G_1 = 0, D_1 = 1$  kada je izlaz I kola "0", što upravo predstavlja signal  $potres_2$ .

I generisanja signala  $potres$  možemo interpretirati u sličnom duhu. Od svih kombinacija signala  $potres_1$  i  $potres_2$ , samo jedna predstavlja stanje gde nije bilo potresa ( $potres_1 = 0, potres_2 = 0$ ). Ovu kombinaciju smo izdvojili od ostalih ILI kolom, i za nju je izlaz ILI kola "0", dok je za ostale kombinacije izlaz "1", što tačno odgovara signalu  $potres$ .

Signal  $kvar$  smo generisali tako što smo od svih kombinacija ulaznih signala izdvojili dve kombinacije, tj. korišćenjem EKSILI (XOR) kola. EKSILI kolo na izlazu daje "1" ako ulazni signali imaju različite vrednosti, odnosno "0" ako ulazni signali imaju istu vrednost.

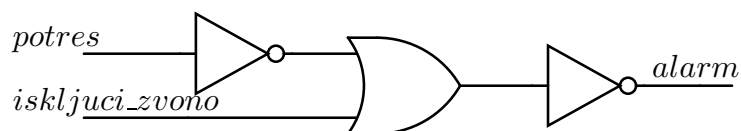
Jedna od nekada korisnih interpretacija I, ILI i EKSILI kola proističe iz jednostavnih svojstava ovih kola prikazanih slikom 1.5.



Slika 1.5: Interpretacije I, ILI i EKSILI logičkih kola

Možemo interpretirati ove slike u smislu da I i ILI kola mogu da "propuštaju" ili "ne propuštaju" ulazni signal kontrolisanjem drugog ulaza kola. Npr. ukoliko na jedan ulaz I kola dovedemo "1", kao da drugi ulaz I kola direktno propuštamo na izlaz. Ukoliko na ulaz I kola dovedemo "0", izlaz je uvek "0" bez obzira na drugi ulaz. Na ovaj način možemo interpretirati kako smo generisali signal  $alarm$ : propusti signal  $potres$  samo ako nije  $iskljuceno\_zvono$ , inače neka je  $alarm = 0$ .

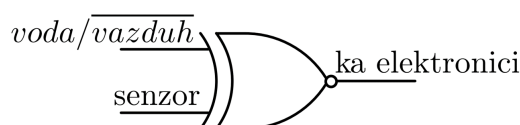
Mogli smo da iskoristimo i ILI kolo. Tada bismo signalom  $iskljuceno\_zvono$  direktno doveli na ILI kolo jer u slučaju da je  $iskljuceno\_zvono = 0$  želimo da dozvolimo propagaciju signala. Međutim, kako u slučaju zabrane propuštanja signala na izlazu ILI kola je stalno "1", moramo invertovati izlazni signal ILI kola, kako  $alarm$  ne bi bio stalno uključen. Zbog toga moramo invertovati i drugi ulaz ILI kola, kako bi "poništili" invertovanje invertora na izlazu ILI kola. Time dobijamo drugi način na koji smo mogli da generišemo signal  $alarm$  prikazan na slici 1.6.



Slika 1.6: Alternativan način generisanja signala *alarm* pomoću ILI kola

Sa druge strane, EKSILI kolo omogućava propuštanje odnosno invertovanje ulaznog signala. Npr. ukoliko na jedan ulaz EKSILI kola dovedemo "0", kao da drugi ulaz EKSILI kola direktno propuštamo na izlaz. Ukoliko na ulaz EKSILI kola dovedemo "1", na izlazu se generiše komplement drugog ulaznog signala.

Iskoristićemo ovu osobinu EKSILI logičkog kola da bismo omogućili korišćenje i senzora koji daju "1" kada su u vodi i senzora koji daju "1" kada su u vazduhu, bez izmene ostatka elektronike. Ukoliko signal  $voda/\overline{vazduh}$  ukazuje na to da li senzor daje "1" kada je u vodi ( $voda/\overline{vazduh} = 1$ ) ili u vazduhu ( $voda/\overline{vazduh} = 0$ ), dodavanjem kola sa slike 1.7 na izlaz senzora dobija se tražena funkcionalnost.



Slika 1.7: Kolo koje se vezuje na izlaz senzora da bi se omogućilo da mogu da se koriste i senzori koji daju "1" u vodi i senzori koji daju "1" u vazduhu

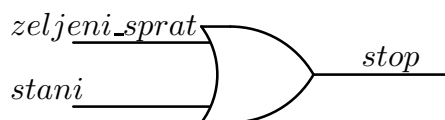
## Zadatak 2

Banka ima 5 spratova i moderan lift. Kako bi stigao do željenog sprata, lift pamti na koji sprat je potrebno otići u vidu signala  $ZS_2$ ,  $ZS_1$  i  $ZS_0$  koji sadrže šifrovani kôd sprata na koji se ide (npr. za sprat 2. je  $ZS_2ZS_1ZS_0 = 110$ , za sprat 5. je  $ZS_2ZS_1ZS_0 = 001, \dots$ ). Na svakom spratu je ugraviran kôd sprata koji lift nakon prolaska pored tog sprata očitava i postavlja vrednosti signala  $TS_2$ ,  $TS_1$  i  $TS_0$  na kôd trenutnog sprata.

Potrebno je generisati signal *stop* kojim se lift zaustavlja nakon što dođe do željenog sprata (lift se zaustavlja ako je  $stop = 1$ ). Lift je takođe potrebno zaustaviti ukoliko je neko pritisnuo dugme u slučaju opasnosti i time postavio signal *stani* na logičku vrednost "1".

## REŠENJE:

Pretpostavimo da imamo signal *zeljeni\_sprat* koji ima vrednost "1" ako se lift nalazi na željenom spratu. Tada je potrebno generisati signal *stop* ako smo na željenom spratu ili ukoliko je aktivan signal *stani* (slika 2.1).

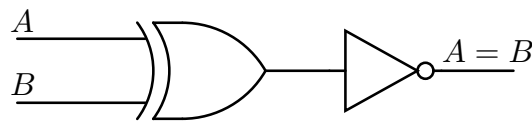


Slika 2.1: Generisanje signala *stop*

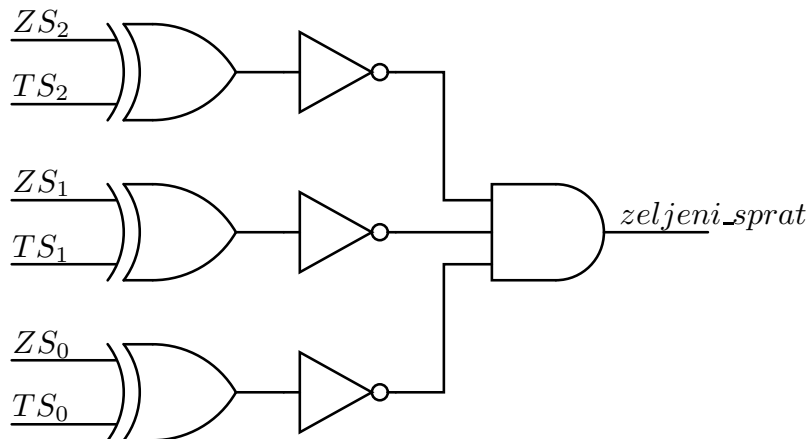
Signal *zeljeni\_sprat* treba da ima vrednost "1" samo ako smo na željenom spratu, odnosno ako je kombinacija signala  $ZS_2$ ,  $ZS_1$ ,  $ZS_0$  identična kombinaciji  $TS_2$ ,  $TS_1$ ,  $TS_0$ , odnosno ukoliko je  $ZS_i = TS_i$  za  $i = 0, 1, 2$ .

Za proveru da li su dva signala jednaka možemo iskoristiti dvoulazno EXILI (XOR) kolo. Ovo kolo daje na svom izlazu vrednost "0" ukoliko su ulazni signali isti, odnosno "1" ukoliko su

različiti. Dakle kolo na slici 2.2 se može koristiti da bismo uporedili dva signala. Realizacija signala *zeljeni\_sprat* prikazana je na slici 2.3.

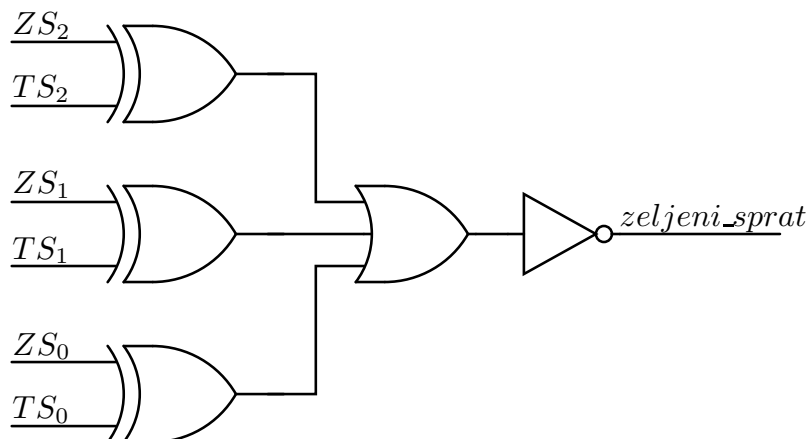


Slika 2.2: Poređenje na jednakost signala



Slika 2.3: Realizacija signala *zeljeni\_sprat*

Ukoliko bismo se vodili logikom da nismo na željenom spratu ukoliko nije  $ZS_i = TS_i$  za bilo koje  $i = 0, 1, 2$ , dobili bismo drugu realizaciju prikazanu na slici 2.4.



Slika 2.4: Alternativna realizacija signala *zeljeni\_sprat*

### Zadatak 3

Kako bi lift u banci bio što efikasniji i opsluživao sve spratove jednako dobro, lift menja smer kretanja samo kada je to neophodno, odnosno onda kada više nema nijedan zahtev u tom smeru kretanja. U cilju sprovođenja ovog algoritma, liftu je potrebna informacija koji je najviši (ili najniži) od svih spratova sa koga postoji zahtev za lift. Signal *gore/dole* ima logičku vrednost "1" ukoliko se lift kreće na gore, odnosno "0" ukoliko se lift kreće na dole. Ukoliko se lift kreće na dole, potrebna je informacija o najvišem spratu sa koga postoji zahtev. Ukoliko se lift kreće na gore, potrebna je informacija o najnižem spratu sa koga postoji zahtev. Imajući

na raspolaganju signale *gore/dole*,  $Z_5$ ,  $Z_4$ ,  $Z_3$ ,  $Z_2$  i  $Z_1$  (gde  $Z_i = 1$  znači da postoji zahtev sa sprata  $i$ ), napraviti signale  $SS_5$ ,  $SS_4$ ,  $SS_3$ ,  $SS_2$  i  $SS_1$  koji predstavljaju sledeći sprat na koji treba otići (gde  $SS_i = 1$  znači da je sledeći sprat na koji se staje sprat  $i$ , u tom slučaju je  $SS_j = 0$  za sve  $j \neq i$ ).

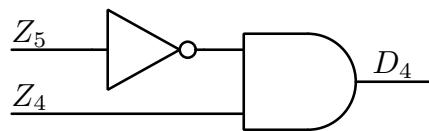
### REŠENJE:

Svedimo problem na manje celine. Napravićemo signale  $D_5$ ,  $D_4$ ,  $D_3$ ,  $D_2$  i  $D_1$  koji će nositi informaciju o najvišem spratu sa koga postoji zahtev (potrebno kada se lift kreće na dole), signale  $G_5$ ,  $G_4$ ,  $G_3$ ,  $G_2$  i  $G_1$  koji će nositi informaciju o najnižem spratu sa koga postoji zahtev (potrebno kada se lift kreće na gore), a zatim ćemo u zavisnosti od signala *gore/dole* postaviti vrednosti signala  $SS_i$  na  $D_i$  ili na  $G_i$ . Napravimo signale  $D_5$ ,  $D_4$ ,  $D_3$ ,  $D_2$  i  $D_1$  tako da je  $D_i = 1$  ukoliko je  $i$  najviši sprat na kome postoji zahtev za lift, i  $D_i = 0$  inače. Ukoliko postoji zahtev  $Z_5$ , tada je automatski  $D_5 = 1$ .



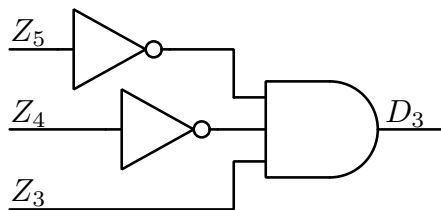
Slika 3.1: Generisanje signala  $D_5$

Sprat 4 je najviši sprat sa koga postoji zahtev (tada je  $D_4 = 1$ ), samo ukoliko postoji zahtev sa tog sprata ( $Z_4 = 1$ ) i ne postoji zahtev sa sprata 5 ( $Z_5 = 0$ ).



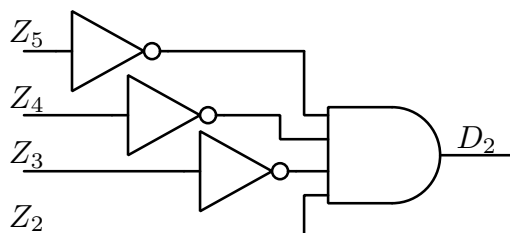
Slika 3.2: Generisanje signala  $D_4$

Sprat 3 je najviši sprat sa koga postoji zahtev (tada je  $D_3 = 1$ ), samo ukoliko postoji zahtev sa tog sprata ( $Z_3 = 1$ ) i ne postoji zahtev sa sprata 5 ( $Z_5 = 0$ ) i ne postoji zahtev sa sprata 4 ( $Z_4 = 0$ ).

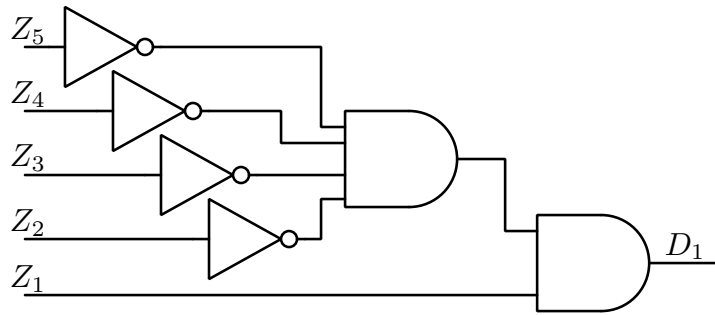


Slika 3.3: Generisanje signala  $D_3$

Slično generišemo i signale  $D_2$  i  $D_1$ .

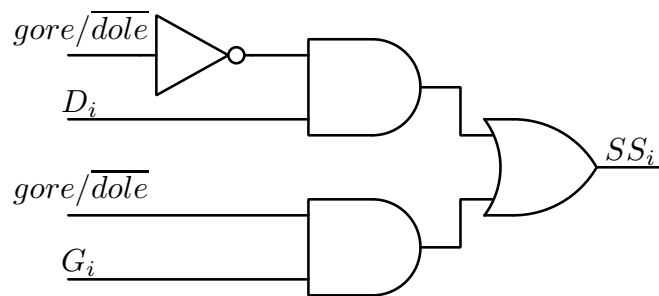


Slika 3.4: Generisanje signala  $D_2$



Slika 3.5: Generisanje signala  $D_1$

Na sličan način možemo generisati i signale  $G_5$ ,  $G_4$ ,  $G_3$ ,  $G_2$  i  $G_1$ , što se ostavlja za vežbu čitaocu. Potrebno je još u zavisnosti od signala  $gore/dole$  postaviti signale  $SS_i$  na vrednosti  $G_i$  ili  $D_i$ .

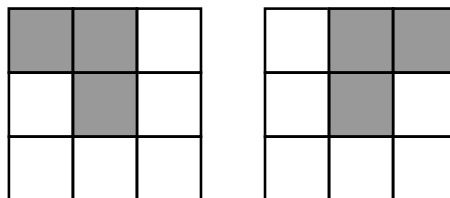


Slika 3.6: Generisanje signala  $SS_i$

Studentu se ostavlja za vežbu da interpretira ovu realizaciju u duhu "propušta"- "ne propušta" svojstava I i ILI kola diskutovanih u rešenju zadatka 1.

#### Zadatak 4

Dijamant se u banci čuva u posebnoj prostoriji zaštićenoj sigurnosnom elektronskom bravom. Bravu moraju da otvore dve osobe istovremeno, osoba koja zna trenutnu šifru i radnik obezbeđenja. Svaka od ove dve osobe mora da pritisne tačno odgovarajuću kombinaciju tastera na tastaturi  $3 \times 3$  tastera. Osoba koja zna šifru treba da pritisne određenu kombinaciju tastera odjednom, dok radnik obezbeđenja treba da pritisne u isto vreme neku drugu kombinaciju tastera na drugoj identičnoj tastaturi.



Slika 4.1: Grupe tastera

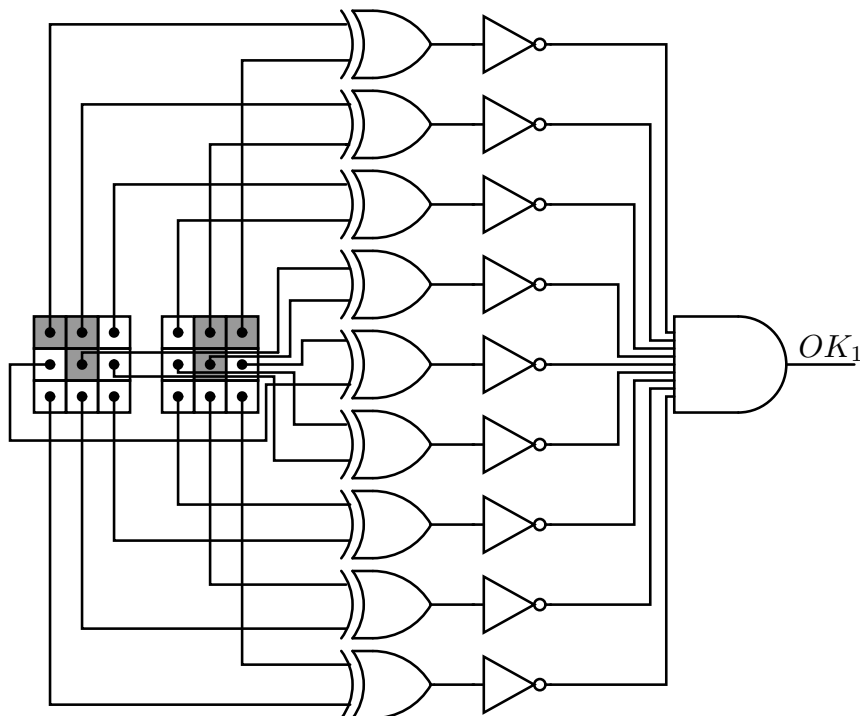
Ova druga kombinacija tastera zavisi od tastera koju pritiska osoba sa šifrom i dana u nedelji. Ponedeljkom je potrebno pritisnuti tastere simetrične šifri u odnosu na vertikalu, utorkom u odnosu na horizontalu, sredom u odnosu na jednu dijagonalu, četvrtkom u odnosu na drugu dijagonalu, petkom centralno simetrično u odnosu na centralni taster, subotom tastere zarotirane u desno za  $90^\circ$ , a nedeljom tastere zarotirane u levo za  $90^\circ$ . Svaki od ovih 18 tastera ukoliko je pritisnut daje logičku vrednost signala "1". Potrebno je napraviti signal *otvori* koji ima

logičku vrednost "1" ukoliko je potrebno otvoriti bravu. Centralni računar sistema obezbeđenja ima signale koji nose informaciju o danu u nedelji.

**REŠENJE:**

Napravićemo poseban signale  $OK_i$  za svaki dan u nedelji koji označavaju da li su tasteri pritisnuti u skladu sa algoritmom, pa ćemo zatim u zavisnosti od dana u nedelji postaviti signal *otvori* na osnovu odgovarajućeg signala  $OK_i$ .

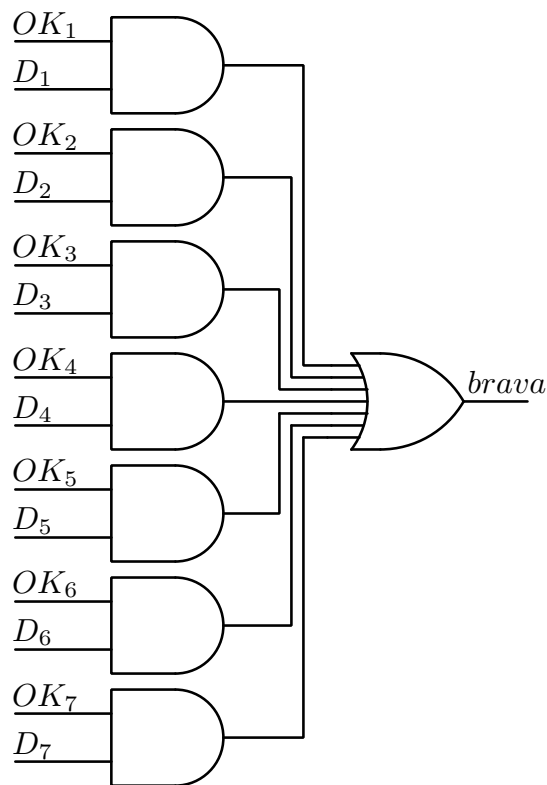
Napravimo signal  $OK_1$  koji će imati vrednost "1" ako je su tasteri na tastaturi pritisnuti u skladu sa algoritmom od ponedeljka, dakle ako su simetrični u odnosu na vertikalu. To znači da je za svaki taster na prvoj tastaturi, taster na drugoj tastaturi koji je simetričan njemu u odnosu na vertikalu i istom stanju kao i taster na prvoj tastaturi. Dakle, imamo 9 parova simetričnih tastera od kojih svaka dva treba da budu u istim stanju. Možemo iskoristiti kolo iz zadatka 2.2 koje ispituje jednakost dva signala. Tada treba generisati signal  $OK_1$  ako i samo ako su u svakom od 9 parova simetričnih tastera oba signala tastera jednaka. Šema je prikazana na slici 4.2. Studentu se ostavlja da napravi odgovarajuće šeme za ostale dane.



Slika 4.2: Generisanje signala  $OK_1$

Pretpostavimo da se informacija o današnjem danu dobija od centralnog računara u vidu signala  $D_1, D_2, D_3, D_4, D_5, D_6$  i  $D_7$ , gde je u danu  $i$  signal  $D_i = 1$  i  $D_j = 0$  za  $j \neq i$ . Tada se signal *brava* može generisati na način prikazan na slici 4.3.



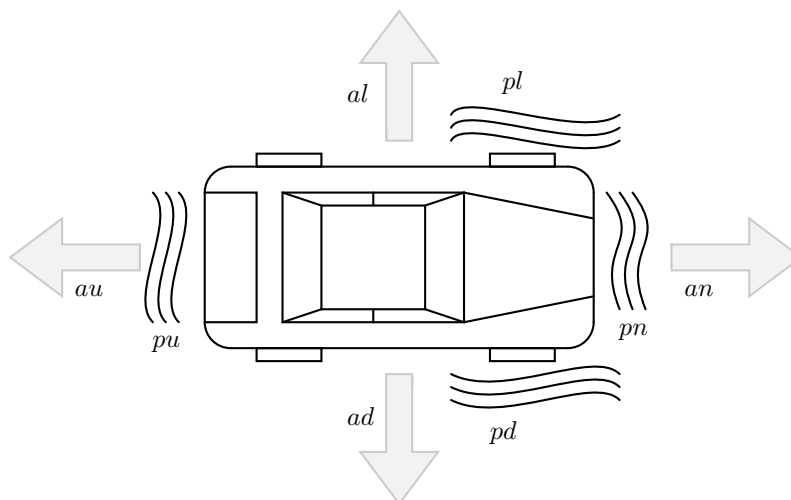


Slika 4.3: Generisanje signala *brava*

Studentu se ostavlja za vežbu da interpretira ovu realizaciju u duhu "propušta"- "ne propušta" svojstava I i ILI kola diskutovanih u rešenju zadatka 1.

### Zadatak 5

Grupa lopova se sprema da ukrade dijamant. Nabavili su igračku-automobil na daljinsko upravljanje koji žele da prilagode svojoj potrebi. Automobil ima upravljačke signale za kretanje napred (*an*), unazad (*au*), levo (*al*) i desno (*ad*). Postavljanjem nekog od ovih signala na vrednost "1", kontroliše se kretanje automobila. Kako bi što manje rizikovali upravljanje automobilom, lopovi su ugradili senzore za prepreke napred (*pn*), unazad (*pu*), levo (*pl*) i desno (*pd*) koji u slučaju prepreke u određenom smeru dobijaju logičku vrednost "1". Kada postoji prepreka treba sprečiti generisanje odgovarajućeg signala za kretanje *ai*.

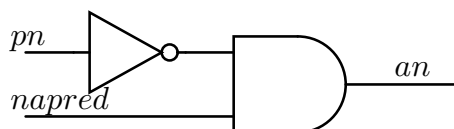


Slika 5.1: Model automobila sa bitnim signalima

Takođe, pošto nisu sigurni da li će automobil voziti ka sebi ili od sebe tokom pljačke, kako se ne bi zbunili sa komandama automobila, dodali su i dugme na daljinskom upravljaču kojim se menja smer kretanja automobila (napred i unazad menjaju uloge, kao i levo i desno). Ukoliko je signal *obrnutto* na logičkoj vrednosti "1", aktivan je suprotan smer kretanja. Tasteri na daljinskom upravljaču generišu signale *un*, *uu*, *ul* i *ud* dajući im vrednost "1" kada su pritisnuti. Pomozite lopovima da naprave logiku koja kontroliše signale za smer kretanja automobila.

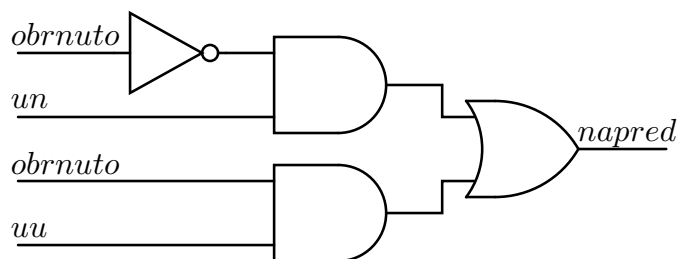
### REŠENJE:

Neka je signal *napred* signal koji se generiše kada se na upravljaču pritisne odgovarajuća komanda. Pošto signal *an* treba da se generiše kada nema prepreke, odnosno kada je signal *pn* = 0, dobija se realizacija kao na slici 5.2.



Slika 5.2: Generisanje signala *an*

Pošto je rečeno da se signalom *obrnutto* kontroliše da li se komande upravljača tumače normalno ili inverzno, onda, u zavisnosti od tog signala, prilikom generisanja signala *napred* treba propustiti odgovarajući signal upravljača, što je prikazano na slici 5.3.



Slika 5.3: Generisanje signala *napred*

Signali za ostale smerove kretanja se generišu na sličan način, što se ostavlja studentu za vežbu.

---

### Zadatak 6

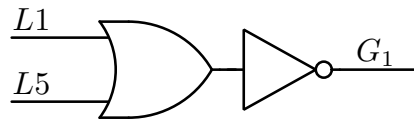
Banka ima lasersko obezbeđenje u sobi sa dijamantom. Senzor lasera daje logičku vrednost "1" ukoliko snop lasera nije prekinut. U prošlosti je banka imala neprijatna iskustva sa česticama prašine i električnim smetnjama koji su često aktivirali alarm. Zbog toga je banka rešila da poboljša sistem tako da se alarm uključuje (*alarm* = 1) samo ako su dva bliska laserska snopa prekinuta. Bliski laserski snopovi obrazuju grupu snopova. Treba generisati signal *alarm* za grupe bliskih snopova (*L1, L5*), (*L1, L2*), (*L2, L3, L4*), (*L3, L5*) i (*L4, L6, L1*).

### REŠENJE:

Ovaj zadatak se jednostavno rešava primenom osobina logičkih ILI kola primećenih u zadatku 1.

Neka za svaku od ovih 5 grupa postoji signal  $G_i$ , koji ukoliko ima vrednost "1" znači da su prekinuti laserski snopovi koji pripadaju grupi  $i$ ,  $i = 1, 2, 3, 4, 5$ .

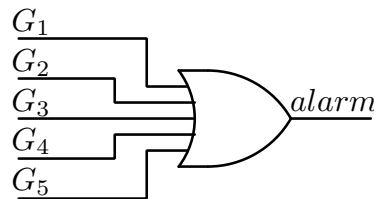
Svaki od tih signala će biti generisan ukoliko su aktivni signali koji kažu da su odgovarajući laserski snopovi prekinuti. Npr. za grupu 1, ukoliko su prekinuti laserski snopovi 1 i 5 (odnosno  $L1 = 0$  i  $L5 = 0$ ), potrebno je generisati signal  $G1$ . Ta realizacija je prikazana na slici 6.1.



Slika 6.1: Generisanje signala  $G_1$

Studentu se prepušta da uradi isto i za ostale grupe.

Konačno, signal *alarm* se generiše ukoliko bilo koja od grupa detektuje da su prekinuti odgovarajući laserski snopovi, tako da se dobija realizacija prikazana na slici 6.2.



Slika 6.2: Generisanje signala *alarm*

### Zadatak 7

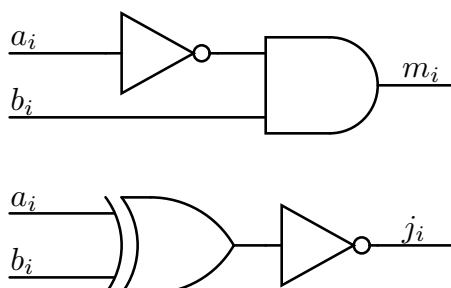
Data su dva neoznačena trocifrena binarna broja  $A = a_2a_1a_0$  i  $B = b_2b_1b_0$ . Generisati signal *manji* ukoliko je broj  $A$  manji od broja  $B$ .

### REŠENJE:

Pošto su brojevi dati u težinskom brojnem sistemu, da bi broj  $A$  bio manji od broja  $B$ , potrebno je da važi:

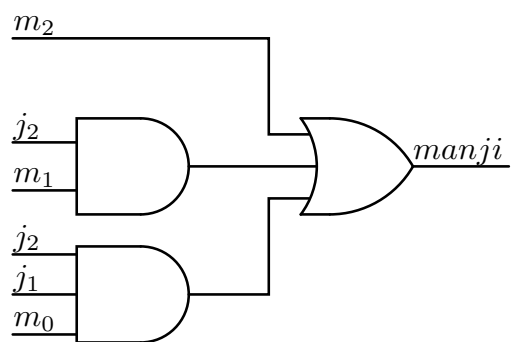
- $a_2 < b_2$ , ili
- $a_2 = b_2$  i  $a_1 < b_1$ , ili
- $a_2 = b_2$  i  $a_1 = b_1$  i  $a_0 < b_0$ .

Brojevi  $A$  i  $B$  su binarni, tako da je  $a_i < b_i$  samo ako je  $a_i = 0$  i  $b_i = 1$ . Koristićemo pomoćni signal  $m_i$  koji je jednak "1" ako je  $a_i < b_i$ , inače je "0" i signal  $j_i$  koji je jednak "1" ako je  $a_i = b_i$ , inače je "0" (slika 7.1).



Slika 7.1: Generisanje signala  $m_i$  i  $j_i$

Odatle se dobija realizacija signala *manji* prikazana na slici 7.2.



Slika 7.2: Generisanje signala *manji*

---