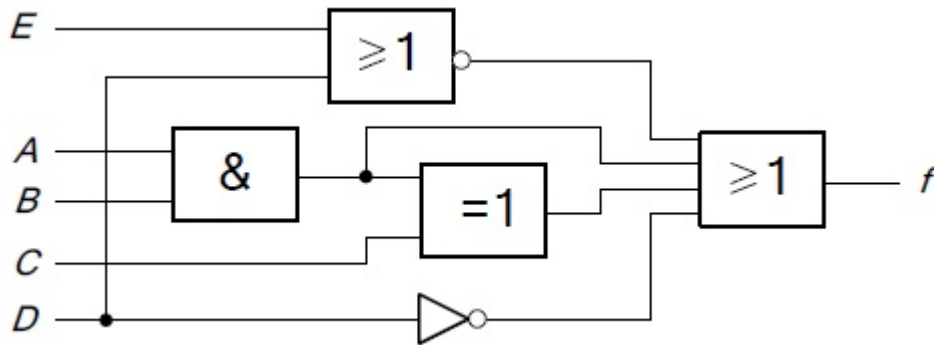


**ZAVRŠNI ISPIT IZ DIGITALNE LOGIKE ZIMSKI ROK
2011./2012.-RJEŠENI ZADATCI**

A.M.

1. Sklop koji ostvaruje funkciju f prikazan je na slici. Uporabom jednog multipleksora 2/1 potrebno je ostvariti sklop koji ostvaruje istu funkciju. Ako se na adresni ulaz multipleksora dovebe varijabla B , kako glasi minimalni zapis rezidualne funkcije koju je potrebno dovesti na podatkovni ulaz d_0 .



Prvo što treba napraviti je očitati funkciju iz sheme na slici.

Druga stvar koju treba napraviti jest odrediti kako ostvariti tu funkciju multipleksorom 2/1.

Dana funkcija je funkcija od 5 varijabli, a kako imamo mux 2/1, jednu varijablu stavimo na selekcijski ulaz a od ostale 4 odredimo dvije rezidualne funkcije koje će ići na podatkovne ulaze. Ukoliko se odlučimo za tablični prikaz kako bi se domogli rezidualnih funkcija, B trebamo staviti na mjesto najvećeg značenja, jer je to selekcijski ulaz. U ovom slučaju nam tablični prikaz nije praktičan jer je tablica velika te se koristimo standardnom Shannonovom dekompozicijom.

Kad očitamo f iz sheme dobivamo $f = (\neg(D * E) + (A * B) + ((A * B) \otimes C) + \bar{D}$

Na d_0 ulaz mux-a dovodimo rezidualnu funkciju $f_{\bar{B}}$ a na ulaz d_1 funkciju f_B

Shannonova dekompozicija nam kazuje $f = \neg B * f(B = 0) + B * f(B = 1)$. Fiksacijom varijable B na konstantnu vrijednost 0 ili 1 eliminira se varijabla B te f ovisi o preostalim varijablama.

U zadatku se traži d_0 , dakle treba nam samo $f(B = 0)$.

Nakon uvrštavanja dobivamo $(\neg(D + E)) + A * 0 + (A * 0) \otimes C + \bar{D}$

Za prvi član iskoristimo De Morganovo pravilo, za drugi i treći pravilo dominacije 0 za and, pa imamo

$\neg D * \neg E + C + \neg D$. ovaj zapis je točan, ali nije minimalan. Kako bi minimalizirali izraz možemo iskoristiti pravilo apsorpcije ili izlučiti $\neg D$ iz prvog i trećeg člana pa dobivamo $\neg D * (\neg E + 1) = \neg D * 1 = \neg D$.

Dakle, minimalni zapis tražene funkcije je $C + \neg D$

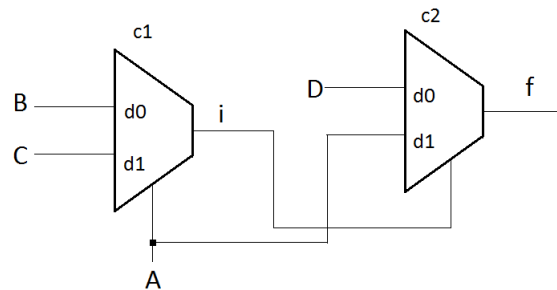
2. Na raspolaganju je multipleksor definiran kao komponenta MUX21 u čijem su sučelju navedeni jednobitni signali d_0, d_1, s i y (upravo tim redoslijedom). Te se komponente koriste u strukturnom opisu komponente sklop koji ima ulaze A,B,C,D te izlaz f. U arhitekturi sklopa nalaze se:

```
c1: entity work.mux21 port map(B,C,A,i);
c2: entity work.mux21 port map(D,A,i,f);
```

Utvrđite minimalni zapis funkcije opisanog sklopa u obliku sume produkata. i je interni signal.

Prvo nacrtamo skicu kombinacijskog sklopa, zatim očitamo funkciju, i tek onda možemo odrediti minimalni zapis dane nam funkcije.

Pitanje: Zašto je važan redoslijed varijabli u port map? Zato što je ovakav način pozivanja pozicijski, pa će se varijable pridruživati redom kako su napisane.



Na skici se vidi da je i interni signal (pomoćni)

Odredimo vrijednost signala i: $i = \neg A * B + A * C$

Odredimo vrijednost f: $f = \neg(i) * D + i * A = \neg(\neg A * B + A * C) * D + (\neg A * B + A * C) * A$

Za prvi član koristimo De Morganovo pravilo, pa dobivamo

$$f = (A * \neg C + \neg A * \neg B + \neg B * \neg C) * D + A * C = A * \neg C * D + \neg A * \neg B * D + \neg B * \neg C * D + A * C$$

To ubacujemo u K-tablicu:

		A			
		00	01	11	10
AB/CD	00	0	0	0	0
	01	1	0	1	1
	11	1	0	1	1
	10	0	0	1	1
		B			

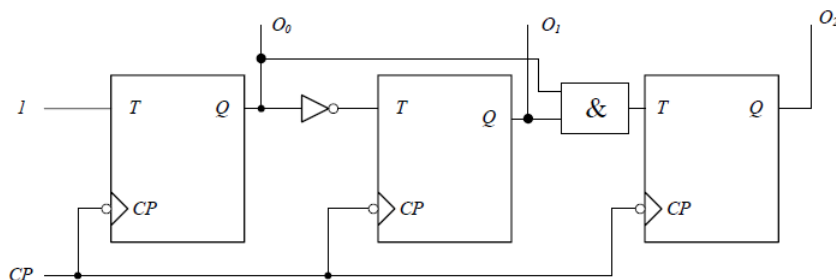
Crveno zaokružen je član $A * D$, zeleno $A * C$ i plavo $\neg B * D$, pa je konačno rješenje:
 $f = A * D + A * C + \neg B * D$.

3. Koji je minimalni dekodер dovoljan kako bismo jednim takvim ostvarili funkciju $f(A, B, C, D, E) = m(0, 11, 15, 16, 27, 31)$

Minimalni dekodер znači da trebamo prvo minimizirati funkciju. To je trivić al evo kratke Quine-Mc Cluskey minimizacije, A nije bitan, te se tako funkcija svodi na 4 varijable, pa je dostatan dek 4/16.

	A	B	C	D	E	iskorišten		A	B	C	D	E
m0	0	0	0	0	0	da	0,16	-	0	0	0	0
m16	1	0	0	0	0	da	15,31	-	1	1	1	1
m11	0	1	0	1	1	da	11,27	-	1	0	1	1
m27	1	1	0	1	1	da	15,11	0	1	-	1	1
m15	0	1	1	1	1	da	27,31	0	1	-	1	1
m31	1	1	1	1	1	da	11,15,27,31	-	1	-	1	1

4. Izvedba nekog automata prikazana je na slici. Izlazi automata su $O_2O_1O_0$. Utvrdite ciklus u kojem se mijenjaju izlazi tog automata. Jedan njegov dio je:



Ovdje je važno primijetiti da je sklop u pitanju sekvencijcki, a ne kombinacijski, te da ima 3 različita bita stanja jer nam treba za način na koji sklop mijenja trenutno stanje. Zatim treba primijetiti kakva je veza bitova stanja i izlaza sekvencijskog sklopa. U ovom slučaju su bitovi stanja direktno izvučeni van, bez dekodera.

Ukupno je moguće 8 stanja. Ispisati ćemo sva ta stanja i za svako odrediti sljedeće. U tablicu ćemo staviti temeljem čega se dobiva sljedeće stanje, tko ga generira-znači bistabili-što se dovodi na ulaze bistabila, što sve utječe na ulaze bistabila. Sljedeće stanje je određeno sadašnjim te pobudom na bistabilu.

$$T_0 = 1, T_1 = \neg Q_0, T_2 = Q_0 * Q_1$$

TRENUTNO STANJE			POBUDA			IDUĆE STANJE		
Q2	Q1	Q0	T2	T1	T0	Q2	Q1	Q0
0	0	0	0	1	1	0	1	1
0	0	1	0	0	1	0	0	0
0	1	0	0	1	1	0	0	1
0	1	1	1	0	1	1	1	0
1	0	0	0	1	1	1	1	1
1	0	1	0	0	1	1	0	0
1	1	0	0	1	1	1	0	1
1	1	1	1	0	1	0	1	0

Općeniti koraci za rješavanje:

1. Raspisati sva moguća stanja
2. Odrediti što se dovodi na ulaze bistabila
3. Sljedeće stanje
4. Čitanje stanja: u ovom slučaju $0 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 7 \rightarrow 2 \rightarrow 1$.

Iz ovog lagano vidimo da nam je sekvenca 365 rješenje.

Ako se doda dodatno kodiranje izlaza, postupak je isti, samo na kraju treba podatke provući kroz koder (na osnovi trenutnog stanja, naravno). Tu je bitno upamtiti da se izlazi mogu ponavljati u jednom ciklusu, ali ne i stanja!!!

5. Za automat prikazan na slici u zadatku 4 utvrdite maksimalnu frekvenciju rada. Kašnjenje logičkog sklopa I je 10ns, invertora 5ns, vrijeme postavljanja bistabila iznosi 20ns, vrijeme kašnjenja bistabila iznosi 30ns a vrijeme pridržavanja bistabila iznosi 18ns. Frekvencije su u odgovorima zaokružene na jednu decimalui navedene su u MHz.

Prvo je potrebno primijetiti da su bistabili sinkroni i reagiraju na padajući brid. To se vidi po trokutiću i kružiću - krug predstavlja negativnu vrijednost, a trokut brid, ergo padajući brid. Da bi odredili frekvenciju još nam je potrebno znati vremenska ograničenja bistabila (vrijeme potrebno da bistabil pravilno radi)

Vrijeme postavljanja, t_{setup} - od trenutka nailaska brida unazad toliko vremena ulaz u bistabil mora biti stabilan= vrijeme prije nailaska brida u kojem pobuda mora biti stalna.

Vrijeme pridržavanja (zadržavanja) - da bi osigurali da bistabil stvarno reagira na pobudu, ona mora biti još toliko vremena stabilna nakon brida.

Uzrok tim vremenima su nesavršeni bridovi koji pak nastaju zbog parazitivnih kapacitivnosti.

t_{DB} je vrijeme kašnjenja bistabila - nakon koliko vremena od nailaska brida će dobra vrijednost biti na izlazu=toliko vremena treba pričekati da se dovede idući padajući brid.

Za B_0 nema t_{setup} jer je T_0 stalan (1). Jedino ograničenje za mogućnost idućeg CP je t_{DB} .

Za B_1 ograničenje je $t_{DB} + t_{Dne} + t_{setup}$

Za sustav B_1B_0 treba čekati $t_{DB} + t_{Dne} + t_{setup}$ - max. od dva vremena, kako bi se zadovoljio najsporiiji bistabil

Za B_2 ograničenje je $t_{DB} + t_{Di} + t_{setup}$

Za sustav $B_2B_1B_0$ treba čekati $t_{DB} + t_{Di} + t_{setup} = 60ns$, max. od tri vremena.

Frekvencija je inverz tog vremena, dakle 16,7 MHz.

6. Neki automat s tri stanja izveden je uporabom 3 bistabila tipa D. Automat ima ulaz X te izlaz Y. Stanja su kodirana na sljedeći način: $S_0=001$, $S_1=010$, $S_2=100$. Pri tome su u kodnoj riječi izlazi bistabila navedeni redoslijedom $Q_2Q_1Q_0$. Ima li ovaj automat siguran start? U koje će stanje automat prijeći ako mu je trenutno stanje S_2 a na ulaz X se dovede vrijednost 1? Za opisanu izvedbu vrijedi: $D_2 = \overline{Q_2}\overline{Q_1}Q_0X + \overline{Q_2}Q_1\overline{Q_0}\overline{X}$, $D_1 = \overline{Q_2}\overline{Q_1}Q_0\overline{X} + Q_2\overline{Q_1}\overline{Q_0}X$, $D_0 = Q_2\overline{X} + Q_2Q_0 + Q_1X + Q_1Q_0$, $Y = Q_2 \oplus Q_0$.

Pogledati zadatak 4. za općenite korake.

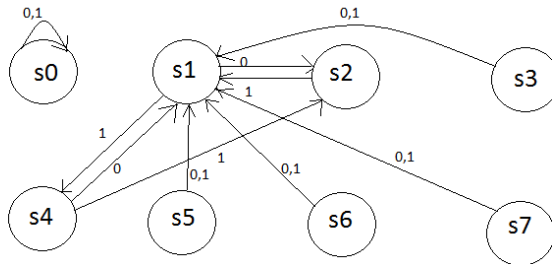
U kojem slučaju automat ima siguran start? Ako postoje barem dva ciklusa onda NEMA jer ne znamo koji je ciklus pravi, a kad se nađemo u jednom, nikako ne možemo prijeći u drugi.

Kontruirajmo tablicu s trenutnim stanjima, pobudama (vidi jednadžbe) i sljedećim stanjima.

$$D_2 = \sum m(3,4), D_1 = \sum m(2,9), D_0 = \sum m(5,6,7,8,10,11,12,13,14,15)$$

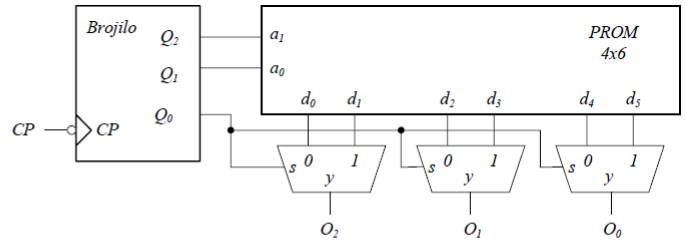
Trenutno stanje			X	Pobuda			Iduće stanje		
Q2	Q1	Q0		D2	D1	D0	Q2	Q1	Q0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	1	0	0	1	0
0	0	1	1	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	1	0	0	1	0	0	1
0	1	1	0	0	0	1	0	0	1
0	1	1	1	1	0	0	1	0	1
1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	0
1	0	1	0	0	0	1	0	0	1
1	0	1	1	0	0	1	0	0	1
1	1	0	0	0	0	1	0	0	1
1	1	0	1	0	0	1	0	0	1
1	1	1	0	0	0	1	0	0	1
1	1	1	1	1	0	0	1	0	1

Iz tablice sad radimo dijagram stanja, ali odmah možemo primijetiti kako stanje S_0 ide samo u sebe, pa smo sigurno da postoje minimalno dva ciklusa, pa sigurnog starta nema. Ipak, evo dijagrama prijelaza stanja (izlaze nisam pisala, nisu potrebni).



Sad iz tablice ili dijagrama vidimo da S_2 za $X = 1$ ide u stanje S_1 . Dakle, odgovor je nema, S_1 .

7. Uporabom sklopa koji se sastoji od binarnog brojala unaprijed te memorije (vidi sliku) potrebno je ostvariti sklop koji na izlazu ciklički generira slijed 7,3,5,1,2,4,0,6. Što je potrebno upisati u memoriju na lokaciju 1? Po uključanju na napajanje binarno brojilo postavit će se u stanje 0 i tada na izlazu čitavog sklopa treba biti 7. Traženi sadržaj memorije u odgovorima je ispisano u oktalnom zapisu. U svim oznakama veći indeks predstavlja bit veće težine.



Nad automatom nemamo kontrolu, mijenja se na padajući brid CP.

Stanja idu $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5 \rightarrow S_6 \rightarrow S_7 \rightarrow S_0$.

A želimo da na izlazi budu $7 \rightarrow 3 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 0 \rightarrow 6 \rightarrow 0$.

Sad treba smisliti samo koder jer imamo dio koji vrti stanja. Ulazi mux 2/1 dobivaju izlaze PROM 4x6 (4 adrese po 6 bitova), a upravljački za mux je Q_0 , dok Q_1 i Q_2 određuju adresu PROM-a.

Tako je za S_0 ili 000 $a_1a_0 = 00$ te je adresirana 0-ta lokacija, tj. gledamo 0-ti redak. Mux propušta $d_0d_2d_4 = 111$ pa to treba biti upisano u PROM.

Za S_1 ili 001 slijedi $a_1a_0 = 01$ te je adresirana 1-ta lokacija, tj. gledamo 1-ti redak. Mux propušta $d_1d_3d_5 = 011$ pa to treba biti upisano u PROM.

itd...

Ovako izgleda PROM:

	d0	d1	d2	d3	d4	d5
0	1	0	1	1	1	1
1	1	0	0	0	1	1
2	0	1	1	0	0	0
3	0	1	0	1	0	0

napomena: d_5 je bit s najvećom težinom.

Treba nam adresa 1, tj. drugi redak pa čitamo: 110001

Za prebacivanje u oktalni sustav grupirano po 3 znamenke počev od najznačajnijeg bita, pa dobivamo da je rješenje 61_8

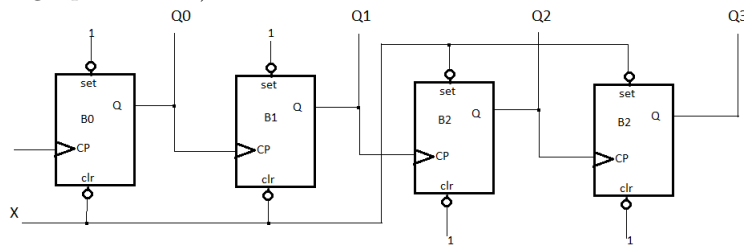
kratka napomena: da zamijenimo a_1 i a_0 samo bi se zamijenili 2. i 3. redak.

8. 4-bitno asinkrono binarno brojilo izvedeno je kao klasično asinkrono binarno brojilo bistabilima tipa T s dodatnim ulazima za postavljanje i brisanje koji djeluju s logičkom nulom. Ulazi za postavljanje bistabila B_0 i B_1 te ulazi za brisanje bistabila B_2 i B_3 spojeni su na logičko 1. Preostali ulazi za postavljanje i brisanje spojeni su zajedno i njima upravlja signal X. Nacrtajte ovaj sklop! Koju Booleovu funkciju treba ostvarivati sklop koji generira signal X ako se želi dobiti asinkrono brojilo koje broji u ciklusu s 12 stanja?

4-bitno asinkrono binarno brojilo ima 16 stanja kroz koja prolazi ciklički

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 0$

Ako se želi ostvariti brojilo od 12 stanja, 12 stanje bi trebalo biti takvo da signal X djeluje na taj način da se brojilo vrati u početno stanje, koje ne mora biti 0000. Aktiviranjem ulaza clr stanje bistabila prelazi u 0, a aktiviranjem ulaza set stanje bistabila prelazi u 1. Budući da su ulazi za postavljanje bistabila B_0 i B_1 te ulazi za brisanje bistabila B_2 i B_3 trajno onemogućeni jer su spojeni na logičku 1, povratak u prvo stanje obavlja se preostalim ulazima, koji očito reagiraju na 0, a ne na 1 (krug ispred ulaza!)



Na skici nisu označeni svi izlazi ni ulazi ni tip bistabila jer nam to nije bitno sad. Samo da vidimo gdje i kako djeluje X, koji nam je jedina mogućnost da vratimo brojilo u početno stanje.

Kad je $X = 1$ ništa se ne događa, brojilo radi normalno, a kad je $X = 0$ B_0 i B_1 se brišu a B_2 i B_3 se postavljaju na 1, dakle kad se ciklus prelama ulazi u stanje 1100. U zadatku kaže da brojilo treba imati 12 stanja dakle zadnje stabilno stanje je S_8 (1000). Znači da je $X = 0$ samo za stanje 8, tj. da je $X = M_8$. (može se dobiti i invertiranjem minterma8, pa primjenom De Morganovog pravila)

$$X = \neg Q_3 + Q_2 + Q_1 + Q_0$$

9. Na raspolaganju je bistabil tipa T . Njegovom uporabom treba ostvariti bistabil čija je jednažba promjene stanja $Q_{n+1} = \bar{A} + Q_n B$. Što je potrebno dovesti na ulaz T ?

Krećemo u projektiranje na isti način kao i prije- prvo gledamo što želimo dobiti, a zatim što imamo na raspolaganju.

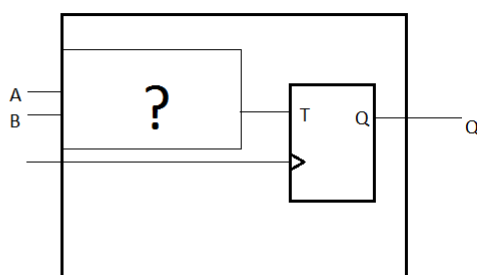
Što želimo dobiti? Bistabil koji funkcionira na sljedeći način $Q_{n+1} = \neg A + Q_n * B$

Što imamo na raspolaganju? T -bistabil.

Sad se pitamo što dovesti na T -bistabil kako bi ostvarili traženo, kakav kombinacijski sklop je potreban. Tu se naravno treba podsjetiti jednažbe stanja T bistabila

$$Q_{n+1} = \neg T * Q_n + T * \neg Q_n$$

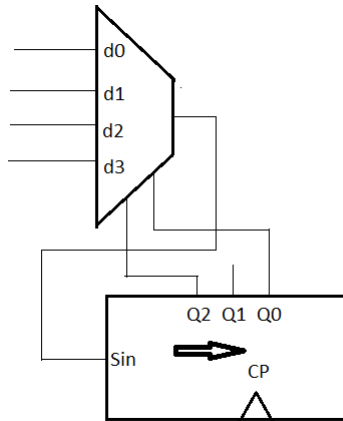
ŠTO ŽELIMO				IMAMO
Q	A	B	Q(n+1)	T
0	0	0	1	1
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0



Iz tablice čitamo $T = m(0, 1, 6)$, tj. $T = \neg Q_n * \neg A + Q_n * A * \neg B$

10. Trobitni posmačni registar ima izlaze $Q_2Q_1Q_0$. Prilikom posmaka, podatak doveden na ulaz S_{in} upisuje se na mjesto Q_2 . Tim ulazom upravlja multipleksor 4/1 na čije je adresne ulaze spojeno $a_1=Q_2$, $a_0=Q_0$. Nacrtajte shemu ovog sklopa. Odredite što je potrebno dovesti na podatkovne ulaze d_0 i d_1 multipleksora kako bi se na izlazima posmačnog registra dobio ciklus 0,4,2,5,6,7,3,1.

Nacrtajte shemu sklopa da ga prepoznate. Pa evo.



Sad radimo tablicu sa svim mogućim stanjima i po zadanom ciklusu(0,4,2,5,6,7,3,1) imamo sljedeća stanja. Prema tome ćemo odrediti S_{in} - to je de facto Q_2 u sljedećem stanju-pogledajte u kojem smjeru se dodaje S_{in} ! Tako S_{in} postaje Q_2 koji postaje Q_1 koji postaje Q_0 .

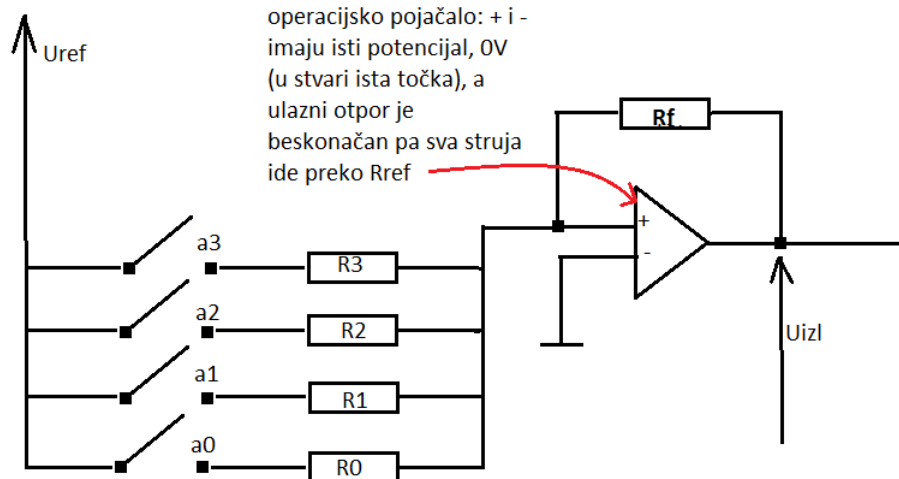
Trenutno stanje			Sljedeće stanje			
Q2	Q1	Q0	Q2	Q1	Q0	S_{in}
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	1	0	1	0	1	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	1	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	1	0

Iz tablice jednostavno čitamo $S_{in} = \neg Q_2 * \neg Q_1 * \neg Q_0 + \neg Q_2 * Q_1 * \neg Q_0 + Q_2 * Q_1$
 Rezidualne funkcije koje ćemo dovesti na ulaze d_0 i d_1 opet određujemo Shannovom dekompozicijom.

Na d_0 se dovodi $S_{in}(Q_2 = 0, Q_0 = 0) = \neg Q_1 + Q_1 = 1$

Na d_1 se dovodi $S_{in}(Q_2 = 0, Q_0 = 1) = 0$

11. 4-bitni težinski D/A pretvornik izveden je s operacijskim pojačalom i težinama $t_3 t_2 t_1 1$, gdje su t_3 , t_2 i t_1 nepoznate težine. Neka je R_0 najveći otpor u težinskoj mreži. Ako je poznato da je $U_{REF} \cdot R_F = 3k V\Omega$, odredite vrijednosti otpora R_0 , R_1 , R_2 i R_3 . Još je poznato da se za ulaz $a_3 a_2 a_1 a_0 = 0001$ dobiva izlaz $-0,1V$, za ulaz 0011 dobiva izlaz $-0,3V$, za ulaz 0100 dobiva izlaz $-0,3V$ te za ulaz 1100 dobiva izlaz $-0,8V$.



Kad je a_0 pritegnuta $I_0 = \frac{U_{ref}}{R_0}$

Kad su a_0 i a_1 pritegnute $I = I_0 + I_1 = \frac{U_{ref}}{R_0} + \frac{U_{ref}}{R_1}$

Općenito, doprinos sklopke a_i ukupnoj struji jest $I_i = \frac{U_{ref}}{R_i}$

Mi imamo 4 komponente struje, $I = I_0 + I_1 + I_2 + I_3$

Sav pad napona se odvija na R_{ref} .

$$0 = I \cdot R_f + U_{izl} \rightarrow U_{izl} = -I \cdot R_f$$

Utjecaji su težine $8 \cdot a_3$, $4 \cdot a_2$, $2 \cdot a_1$, $1 \cdot a_0$

$$I_3 = 2 \cdot I_2 = 4 \cdot I_1 = 8 \cdot I_0$$

Što više težine rastu, to su pripadni otpori sve manji da struja bude veća, pa da U_{izl} bude veći.

Znamo da je doprinos znamenke a_0 je $-0.1V$

U zadatku kaže da je $U_{ref} \cdot R_f = 3kV\Omega$, pa imamo $U_{ref} = \frac{3k}{R_0} = 0.1V \rightarrow R_0 = 30k\Omega$

Općenita formula: $U_{izl} = -U_{ref} \cdot R_f \left(\frac{a_0}{R_0} + \frac{a_1}{R_1} + \frac{a_2}{R_2} + \frac{a_3}{R_3} \right)$

Za $a_3 a_2 a_1 a_0 = 0001$ $U_{ref} = -0.1V \rightarrow R_0 = 30k\Omega$

Za $a_3 a_2 a_1 a_0 = 0011$ $U_{ref} = -0.3V \rightarrow -3k \left(\frac{1}{R_0} + \frac{1}{R_1} \right) = -0.3R_1 = 15k\Omega$ (znamo R_0)

Za $a_3 a_2 a_1 a_0 = 0100$ $U_{ref} = -0.3V \rightarrow R_2 = 10k\Omega$

Za $a_3 a_2 a_1 a_0 = 1100$ $U_{ref} = -0.8V \rightarrow -3k \left(\frac{1}{R_2} + \frac{1}{R_3} \right) = -0.8R_1 = 6k\Omega$ (znamo R_2)

Otpori su redom 30Ω , 15Ω , 10Ω , 6Ω

12. Predajnik i prijemnik razmjenjuju poruke koje sadrže 1 bit informacije i koje su zaštićene Hammingovim kodom uz neparni paritet. Neka je prijemnik s komunikacijskog kanala očitao $y_1y_2y_3$ (uz uobičajen razmještaj zaštitnih i podatkovnih bitova). Neka $s_1(y_1, y_2, y_3)$ predstavlja najznačajniji bit pripadnog sindroma. Kako glasi zapis te funkcije u obliku sume minterma?

Pretpostavljam da znate kodirati Hamminga.

Raspored je $c_1c_2d_1$, a zbog neparnog pariteta slijedi

$$c_1 = c_2 = d_1 \otimes 1 \longrightarrow c_1 = c_2 = \neg d_1$$

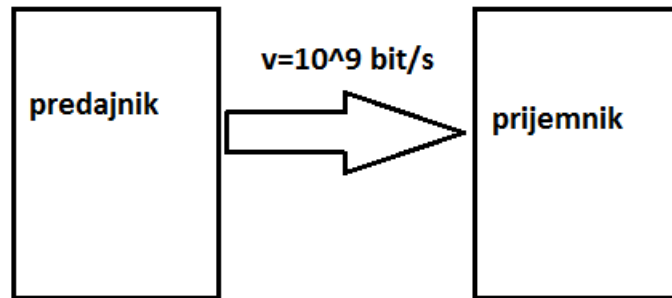
Sindrom je $s_1 = c_{2\text{primljeni}} \otimes c_{2\text{traženi}}$ i $s_0 = c_{1\text{primljeni}} \otimes c_{1\text{traženi}}$

Evo i tablice (sve kombinacije, što imamo, što trebamo):

Primljeno			Traženo		Sindrom	
C1	C2	D1	C2	C1	S1	S0
0	0	0	1	1	1	1
0	0	1	0	0	0	0
0	1	0	1	1	0	1
0	1	1	0	0	1	0
1	0	0	1	1	1	0
1	0	1	0	0	0	1
1	1	0	1	1	0	1
1	1	1	0	0	1	0

Iz tablice očitamo $s_1 = \sum m(0, 3, 4, 7)$

- 13 Predajnik i prijemnik povezani su komunikacijskim kanalom koji u sekundi može prenijeti 10^9 bitova. Kako predajnik svake sekunde generira samo $4 \cdot 10^7$ bitova podataka, inženjeri su odlučili neiskorišteno vrijeme na komunikacijskom kanalu iskoristiti za prijenos zaštitnih bitova te implementirati uporabu koda n -strukog ponavljanja kako bi povećali otpornost na pogreške. Koliko se maksimalno pogrešaka može dogoditi u tako poboljšanom sustavu **da bi postupak ispravljanja i dalje radio korektno?**



Prijemnik generira samo $4 \cdot 10^7 \text{ bit/s}$ a brzina komunikacijskog kanala je 10^9 bit/s . n -struko ponavljanje je najjednostavnija i najneefikasnija zaštita.

Koraci:

1. Koliko puta poruku mogu ponoviti?

Ukupno ima 10^9 bita, originalno $4 \cdot 10^7$ bita, pa je broj ponavljanja $n = \frac{10^9}{4 \cdot 10^7} = 25$.
25 puta predajnik šalje poruku prijemniku.

2. Distanca n -strukog ponavljanja, d_{min} .

3. Može se otkriti $d_{min} - 1$ pogrešaka.

4. Može se ispraviti $\frac{d_{min}-1}{2} = \frac{n-1}{2} = 12$ za naš sluèaj.

Dakle, može se ispraviti 12 grešaka.

primjer. kod 5 - strukog ponavljanja - èitav kod je {00000,11111} pa je distanca koda 5(XOR po bitovima i onda koliko ima jedinica). Treba 3 greške da bi takav sustav zakazao.

napomena. Za sustav u zadatku je minimalno dovoljno 13 grešaka za malfunkciju sustava.

14 Arhitektura kod VHDL opisa sklopa čiji su ulazi A , B i C a izlazi X i Y sadrži sljedeće dvije naredbe:

```
X <= (A AND C) OR (NOT A AND B);
Y <= (B OR C) AND (A OR C);
```

Koja će biti vrijednost izlaza X i Y ako se na ulaze dovede $A='0'$, $B='1'$, $C='U'$?

U VHDL-u signali mogu osim 0 i 1 poprimiti još 7 vrijednosti, jedna od kojih je U (uninitialized). 0 ili 1 ali se ne zna koji.

$x=(0 \text{ and } U) \text{ or } (\text{not } 0 \text{ and } 1) = 0 \text{ or } 1 = 1$

Prvi član smo mogli izračunati zbog dominacije 0 u and.

$y=(1 \text{ or } U) \text{ and } (0 \text{ or } U) = 1 \text{ and } U = U$

Prvi član smo mogli izračunati zbog dominacije 1 u or, drugi nismo jer je 0 neutralni element u or.

napomene. u VHDL-u je not najvećeg prioriteta, a zatim idu and i or (jednakog prioriteta), pa treba paziti na zagrade.

not U=U !!! Jer se ne zna vrijednost. Također, zakoni

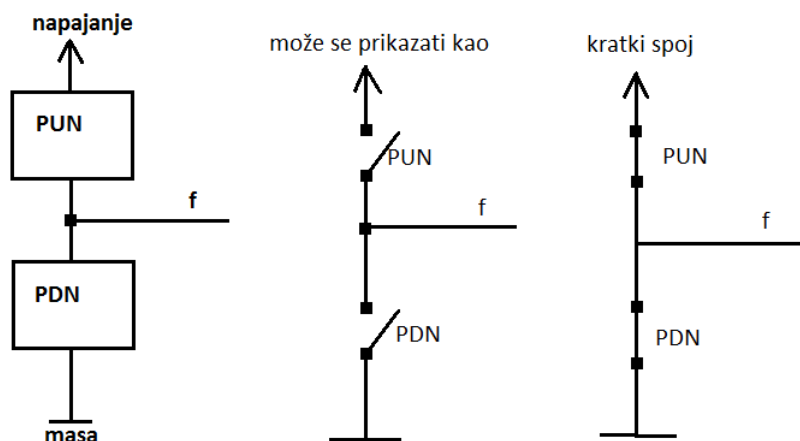
$A \text{ and not } A=0$ i $A \text{ or not } A=1$ vrijede samo u klasičnoj logici. Ako u VHDL-u umjesto A uvrstimo U, rezultat je u oba slučaja U!

15. Čemu je proporcionalna dinamička disipacija snage kod integriranih logičkih sklopova?

Samo se treba sjetiti formule za dinamičku disipaciju snage, $P_{din} = k * C * U^2 * f$
Snaga raste s kvadratom napona i linearno s povećanjem frekvencije.

Ovo sad nije bitno za zadatak al je kratko objašnjenje CMOS-a.

Prisjetimo se CMOS-a na konceptualnoj razini. Idejna shema: Pull-Up-Network za pritezanje na napon napajanja i Pull-Down-Network za pritezanje na masu.



Kad želimo dobiti logičku 1 prvo sklopka je zatvorena, a druga otvorena. Ako želimo 0, obrnuto.

Tim sklopkama upravljaju PUN i PDN ovisno o varijablama koje smo doveli. Problem je što PUN reagira sporije od PDN te se stoga pri prijelazu iz 1 u 0 javlja kratki spoj. Poteći će velika struja koja će disipirati snagu ali je vrijeme dovoljno maleno da izvor ne pregori. U stacionarnom stanju CMOS ništa ne troši, te nema disipacije energije jer nema struje.

16. Koliko nam minimalno treba p -kanalnih tranzistora da bismo u CMOS tehnologiji ostvarili funkciju

$$f(A, B, C, D, E) = (A + B) \cdot (\overline{C} + D \cdot E) ?$$

Komplementi varijabli unaprijed nisu dostupni.

CMOS-om možemo direktno ostvariti funkciju u kojoj su sve varijable komplementirane. U tom slučaju po 1 varijabli se troši 1 p -kanalni za PUN i 1 n -kanalni tranzistor za PDN.

Ajmo prvo malo primjera prije rješavanja zadatka.

$f_1 = \neg A * \neg B(\neg C + \neg D)\neg E$ ima 5 p i 5 n -kanalnih tranzistora, ukupno 10.

$f_2 = A * \neg B(\neg C + \neg D)\neg E$ ne možemo ostvariti direktno, pa uvodimo novu varijablu $X = \neg A \rightarrow A = \neg X$ pa je $f_1 = \neg X * \neg B(\neg C + \neg D)\neg E$ pa bi za implementaciju u CMOS-u trebalo 5 p , 5 n i 10 Σ . ALI! X smo izmislili pa nam treba još invertor(+1 p , +1 n , +2 Σ) za A jer je $A = \text{not } X$, pa je za implementaciju potrebno 5+1 p , 5+1 n tj. Σ 12 tranzistora.

$f_3 = A * B(\neg C + \neg D)\neg E$ ne možemo ostvariti direktno, pa uvodimo dvije nove varijable, $X = \neg A$ i $Y = \neg B$. Sad nam trebaju dva invertora, pa nam za implementaciju treba 5+2 p , 5+2 n , tj. Σ 14 tranzistora.

Za $f_4 = A * B(C + (DE))$ dolazimo do 20 tranzistora.

Postavlja se pitanje može li biti manje tranzistora?

Uzmimo komplement od f_4 , $g = \neg f_4$ (Komplementarna funkcija se radi tako da napravimo dualnu-zamijenimo and i or te logičke 0 i 1, a zatim komplementiramo varijable).

$g = (\neg A + \neg B)\neg C * (\neg D + \neg E)$ pa nam treba 5 p , 5 n , Σ 10. A za f_4 nam treba još invertor pa nam treba još +1 p , +1 n , +2 Σ , ukupno 6 p , 6 n i 12 Σ

Kako odrediti kad je pametnije implementirati funkciju direktno, a kad preko komplementa?

To ovisi o omjeru varijabli sa i bez komplementa. Ako je puno varijabli s komplementom, tada koristimo direktnu implementaciju, a ako je puno varijabli bez komplementa tada koristimo implikaciju preko komplementare funkcije.

Za naš zadatak odabiremo metodu s komplementom:

$$f = (A + B)(\neg C + D * E)$$

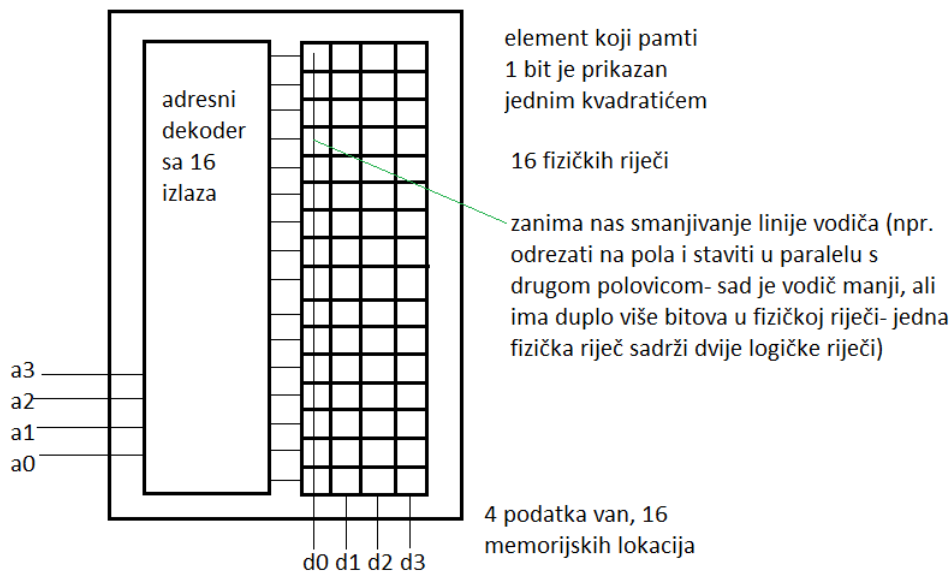
$\neg f = (\neg A * \neg B) + (C * (\neg D + \neg E))$ Treba nam invertor za C , pa je potrebno 6 p , 6 n , Σ 12 tranzistora.

Još nam treba i komplement funkcije, pa nam treba 7 p , 7 n , Σ 14 tranzistora.

17. Memorija organizacije $2 \frac{1}{2} D$ ima 2^9 fizičkih riječi te pristupni multipleksor/demultipleksor s 4 adresna ulaza i 8 izlaza.

Koliki je ukupni kapacitet te memorije u bitovima?

Prvo ćemo objasniti organizaciju memorije. Za primjer uzimamo ROM 16×4 . 16×4 znači da imamo 16 memorijskih lokacija (a time i adresa) a 4 je broj bita koliko svaka memorijska lokacija pamti. Kod 2D organizacije:



Zamislamo da prerežemo liniju vodiča na četvrtine - sad imamo 4 fizičke riječi po 4 logičke riječi, a dekodera $4/16$ nam više ne odgovara (Treba nam $2/4$ na koje dovodimo npr. a_3a_2). To je princip organizacije $2 \frac{1}{2}$. Ali što sad s odabirom logičke riječi? (adresnim bitovima odabire se samo fizička riječ). Za to nam treba mux $4/1$ sa selekcijskim ulazima npr. a_1a_0 . Može se memorija prelamati na proizvoljan broj puta ali je optimalna konfiguracija kvadratna.

Vratimo se na zadatak.

Imamo 2^9 fizičkih riječi pa adresni dekodera ima 9 bitova.

Pristupni mux/dmux (ono naše u $2 \frac{1}{2}$) ima 4 ulaza koji biraju jednu od logičkih riječi kojih ima $2^4 = 16$.

Jedna fizička riječ ima 16 logičkih po 8 bitova (to znamo po broju izlaza iz mux/dmux).

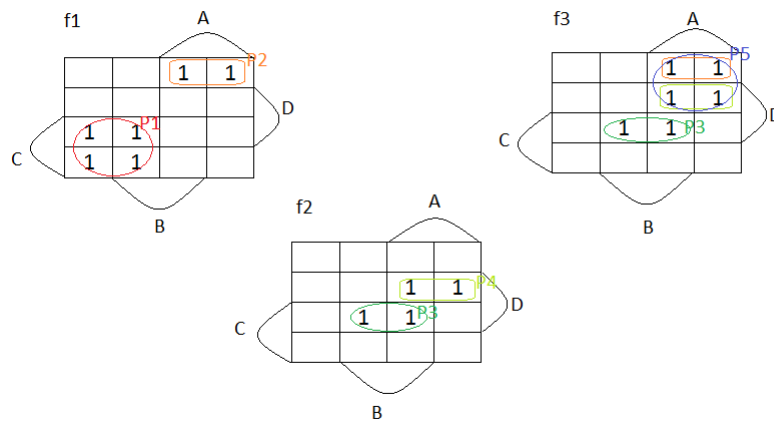
Ukupni kapacitet u bitovima jest $2^9 * 2^4 * 2^3 = 2^{16}$

18. Na raspolaganju je PLA sklop tipa NI-NI, kojim je potrebno u dvije razine logike ostvariti funkcije
 $f_1(A, B, C, D) = \sum m(2, 3, 6, 7, 8, 12)$, $f_2(A, B, C, D) = \sum m(7, 9, 13, 15)$ i
 $f_3(A, B, C, D) = \sum m(7, 8, 9, 12, 13, 15)$. Koje su njegove minimalne dimenzije $m \times n \times k$? m je broj ulaza, n je broj sklopova NI u prvom polju a k broj sklopova NI u drugom polju.

Smijemo li dijeliti produkte za određene funkcije? Da, smijemo kad je u pitanju PLA (programljivo logičko polje) struktura, ali ne kad je u pitanju PAL (poluprogramljivo logičko polje) struktura.

Ideja za rješavanje zadatka: združena minimizacija (možda ne svaku do kraja, ali dijelove dijelimo između više funkcija pa smanjujemo dimenzije strukture).

K-tablice za funkcije:



Iz K-tablica čitamo: $f_1 = p_1 + p_2$, $f_2 = p_3 + p_4$, $f_3 = p_3 + p_5 = p_3 + p_4 + p_2$

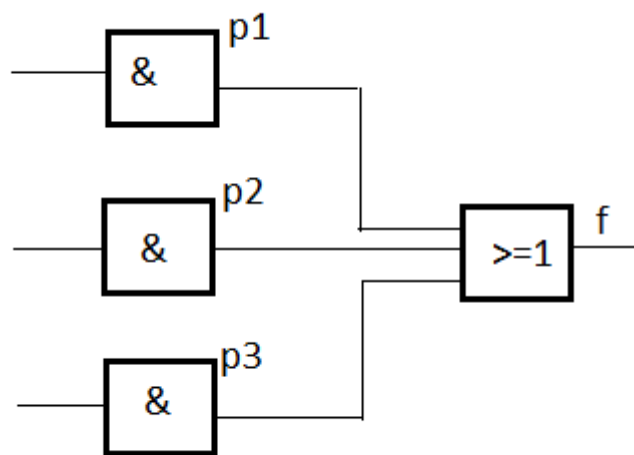
U dimenzijama PLA strukture m je broj varijabli(4), k je broj funkcija (3), a n je broj produkata. Tu vidimo da ako skroz minimiziramo 3. funkciju imamo jedan produkt više, pa PLA nije minimalna, dakle pogrešno je minimizirati jednu po jednu funkciju, treba tražiti zajedničke produkte. Kad tako minimiziramo funkcije, produkata ima 4, pa je $n=4$ te su stoga dimenzije funkcije $4 \times 4 \times 3$.

19. Kojeg je tipa hazard koji može nastati prilikom promjene pobude na jednom od ulaza digitalnog sklopa koji ostvaruje Booleovu funkciju u obliku sume produkata dobivenu Quine-McCluskeyevim postupkom:

- a) dinamički 0-1 hazard
- b) statički 1-hazard
- c) dinamički 0-hazard
- d) statički 0-hazard
- e) dinamički 1-hazard
- f) ništa od navedenoga

Vidimo da je funkcija zadana u obliku sume produkata, pa odmah možemo odbaciti nebuloze o dinamičkim hazardima. Također možemo zaboraviti na koji način smo minimizirali funkciju - to nam je svejedno.

Ovako izgleda sklop:



Je li moguće da se na sva tri izlaza generira 0 za neku pobudu? Da.

Postoji mogućnost da je na točno jednom produktu izlaz 1 (na ostalima 0) te je $f=1$. Postoji mogućnost da je na dva produkta izlaz 1 (na preostalom 0) te je $f=1$.

Što je hazard? Mogućnost privremene promjene izlaza prilikom promjene iz jednog u drugo polje K-tablice.

Statički 0 hazard nije moguć jer su prilikom prebacivanja po K-tablici iz 0 u 0, svi produkti stalno 0.

Statički 1 hazard je moguć dok postoji prijelaz iz 0 u 1 neke varijable čiji komplement se koristi u jednom od produkata. (Ali ne dok se skače unutar jednog od produkata). Isto tako je kod funkcije u obliku produkta sume moguć samo statički 0 hazard.

20. Sklop za izdvojeno generiranje prijenosa generira bitove prijenosa c_0 , c_1 , c_2 i c_3 . Prema kojem algebarskom izraz se generira c_2 ?

Tu vam evo izvoda iz knjige (Peruško, Glavinić - Digitalni sustavi), vrlo je lijep.

Izdvojeno generiranje prijenosa

Zbrajanje istovrsnih bitova ulaznih brojeva obavlja se sklopom za zbrajanje sa sl. 7.42. Važno je još jedanput uočiti da izlazi iz prvog poluzbrajala ne ovise o prijenosu s nižega brojnog mjesta. Za razmatranje koje slijedi prikladno je drugačije označiti te izlaze, tako da se izlaz S' označi sa P , a izlaz C' sa G . Vrijednost G je tzv. *generirajući član* jer će za $G = 1$ biti generiran prijenos na više mjesto bez obzira na prijenos s nižeg brojnog mjesta. Član P je *propagirajući član* jer za $P = 1$, a i $C_{i-1} = 1$ (prijenos s nižega brojnog mjesta), on omogućuje pojavu, tj. *propagiranje* prijenosa na izlaz. Ako se razmatra bilo koji i -ti sklop za zbrajanje tada jednadžbe (7.20) do (7.23) postaju:

$$P_i = A_i \oplus B_i \quad (7.26)$$

$$G_i = A_i B_i \quad (7.27)$$

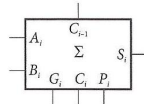
$$S_i = P_i \oplus C_{i-1} \quad (7.28)$$

$$C_i = G_i + P_i \cdot C_{i-1} \quad (7.29)$$

Za četverobitno zbrajalo sa sl. 7.44. vrijedit će:

$$\begin{aligned} C_0 &= G_0 \text{ jer je } C_{-1} = 0 \\ C_1 &= G_1 + G_0 \cdot P_1 \\ C_2 &= G_2 + G_1 \cdot P_2 + G_0 \cdot P_2 \cdot P_1 \\ C_3 &= G_3 + G_2 \cdot P_3 + G_1 \cdot P_3 \cdot P_2 + G_0 \cdot P_3 \cdot P_2 \cdot P_1 \end{aligned} \quad (7.30)$$

Tako definirani kombinaćijski sklop generirat će signale prijenosa istovremeno za sve stupnjeve zbrajala i time ukupno kašnjenje svesti na kašnjenje od $2t_D$ za generiranja G i P te dodatno kašnjenje od $2t_D$ za generiranje prijenosa. Ukupno će dakle kašnjenje biti $4t_D$. Radi realiziranja ove koncepcije potrebno je i potpuno zbrajalo ostvariti s izvodima za G i P . Simbol takvoga sklopa prikazan je na sl. 7.45.



Slika 7.45. Simbol potpunog zbrajala s izvodima G i P

Tvornički se proizvode sklopovi za izdvojeno generiranje prijenosa. Kod takvih se sklopova indeksi ulaznih i izlaznih varijabli često općenito označuju sa $i-x$. U općenitom slučaju ni prijenos s nižega brojnog mjesta neće biti 0. Ako je prijenos s nižeg brojnog mjesta označen sa C_{i-4} tada su jednadžbe za izdvojeno generiranje prijenosa za četverobitno zbrajalo:

$$\begin{aligned} C_{i-3} &= G_{i-3} + G_{i-4} \cdot P_{i-3} \\ C_{i-2} &= G_{i-2} + G_{i-3} \cdot P_{i-2} + G_{i-4} \cdot P_{i-4} \cdot P_{i-3} \\ C_{i-1} &= G_{i-1} + G_{i-2} \cdot P_{i-1} + G_{i-3} \cdot P_{i-3} \cdot P_{i-2} + \\ &\quad + G_{i-4} \cdot P_{i-4} \cdot P_{i-2} \cdot P_{i-3} \end{aligned} \quad (7.31)$$

Sklop za *izdvojeno generiranje prijenosa* definiran je jednadžbama (7.31) i prikazan je na sl. 7.46.

Na sl. 7.47. prikazana je shema sklopa četverobitnog zbrajala izvedenog uporebom sklopa za izdvojeno generiranje prijenosa. Izlazi G' , C' i P' sklopa za izdvojeno generiranje prijenosa služe za generiranje prijenosa kod kaskadiranja sklopova pri izvedbi višebitnih zbrajala.

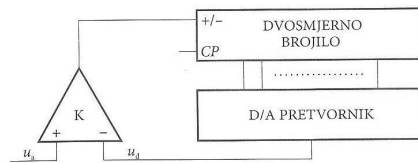
21. U nekom procesu posredno mjerimo neku fizikalnu veličinu pretvorbom u naponski signal. Ako je poznato da se taj napon ne mijenja više od jednog kvanta po periodu signala takta, a želimo taj napon uzorkovati što češće, koji je pretvornik najprikladniji? Želimo dobiti 12-bitni rezultat.

- | | |
|--|---|
| a) Wilkinsonov pretvornik | d) pretvornik sa sukcesivnom aproksimacijom |
| b) brojeći A/D pretvornik | e) težinski D/A pretvornik |
| c) kontinuirano brojeći A/D pretvornik | f) ništa od navedenoga |

Treba shvatiti princip rada svakog od pretvornika pa odabrati. Treba nam A/D pretvornik. Najbolji je kontinuirano brojeći A/D pretvornik. Zašto? (opet malo iz knjige Peruško, Glavinić - Digitalni sustavi)

13.3.5. Slijedni AD-pretvornik

Do sada razmotreni AD-pretvornici obavljali su pretvorbu ulaznoga analognog napona u nekom konačnom vremenskom intervalu nazvanom vrijeme pretvorbe. Pri tome je bilo pretpostavljeno da je ulazni napon konstantan za trajanja procesa pretvorbe. Ako je mjerena

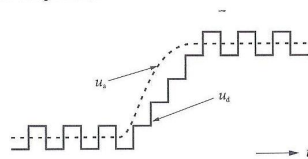


Slika 13.20. Slijedni AD-pretvornik

veličina relativno sporo promjenljiva, vrijeme odziva može se znatno skratiti ako se pretvorba provodi kontinuirano. Takav se pretvornik naziva *slijedni DA-pretvornik* (engl. *tracking DAC*), a izvedba je prikazana na sl. 13.20.

Budući da proces teče kontinuirano, sklop nema posebnog ulaza za start. Može se također zapaziti da je taj sklop modifikacija brojećeg AD-pretvornika razmotrenog u točki 13.3.2. Osnovna razlika je u upotrebi dvosmjernog brojila. Dvosmjerno brojilo (točka 11.7.2) takvo je brojilo koje, ovisno o upravljačkom ulazu, dolazne impulse stanju brojila pribraja ili od njegova stanja oduzima. Pri prvome uključivanju brojilo može biti u bilo kojem stanju. Ako je $u_x > u_d$, na izlazu komparatora bit će vrijednost 1 i brojilo će pribrajati dolazne CP-impulse. Umjesto CP-impulsa može postojati poseban generator impulsa, kao što je to moguće i kod drugih vrsta pretvornika.

Nakon što u_d dosegne razinu napona u_x , proces se nastavlja tako što izlaz iz DA-pretvornika *oscilira* oko vrijednosti analognog napona. Vremenski dijagram procesa pretvorbe prikazan je na sl. 13.21. Do oscilacija dolazi zbog toga što u_d mora premašiti vrijednost u_x , makar za vrlo malen iznos, da bi došlo do promjene izlaza iz komparatora. Ta promjena, međutim, znači da sljedeći impuls treba oduzeti i proces se tako nastavlja. Ako nakon takva stanja dođe do promjene ulaznog napona, odziv će sklopa biti kao na slici. Kad nakon stacionarnog stanja osciliranja u_x počne rasti i kad pri kraju stepenice bude veći od u_d , sljedeći će impuls biti pribrojen. U svim koracima u kojima taj uvjet bude ispunjen nastavit će se i pribiranje sve dok se opet ne postigne neko novo stacionarno stanje. Budući da u ovoj sklopu brojilo neprestano radi, takav se pretvornik katkad naziva i *kontinuirano brojeći AD-pretvornik* (engl. *continuous-digital ramp DAC*).



Slika 13.21. Odziv slijednog AD-pretvornika

22. Modul za digitalnu aritmetiku u nekom sustavu radi sa znamenkama u bazi 4, i pri tome koristi kôd $0 \equiv 01$, $1 \equiv 11$, $2 \equiv 10$, $3 \equiv 00$. Neka su ulazi sklopa koji u tom modulu računa 3-komplement znamenke označeni x_1x_0 a izlazi y_1y_0 . Vrijedi:

a) $y_1 = x_1, y_0 = \bar{x}_0$

c) $y_1 = x_1, y_0 = x_0$

e) $y_1 = \bar{x}_0, y_0 = x_1$

b) $y_1 = \bar{x}_1, y_0 = x_0$

d) $y_1 = \bar{x}_1, y_0 = \bar{x}_0$

f) ništa od navedenoga

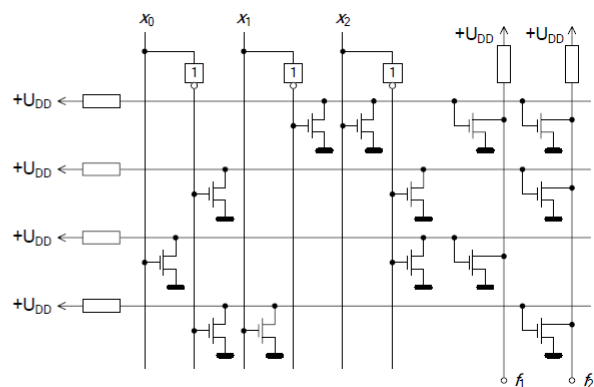
Prvo je bitno zapaziti da se komplement ne određuje u binarnom sustavu nego u sustavu s bazom 4.

Što imamo (pobuda)		U bazi 4	Kompl. U bazi 4	Što nam treba(izlazi)	
X1	X0			Y1	Y0
0	0	3	0	0	1
0	1	0	3	0	0
1	0	2	1	1	1
1	1	1	2	1	0

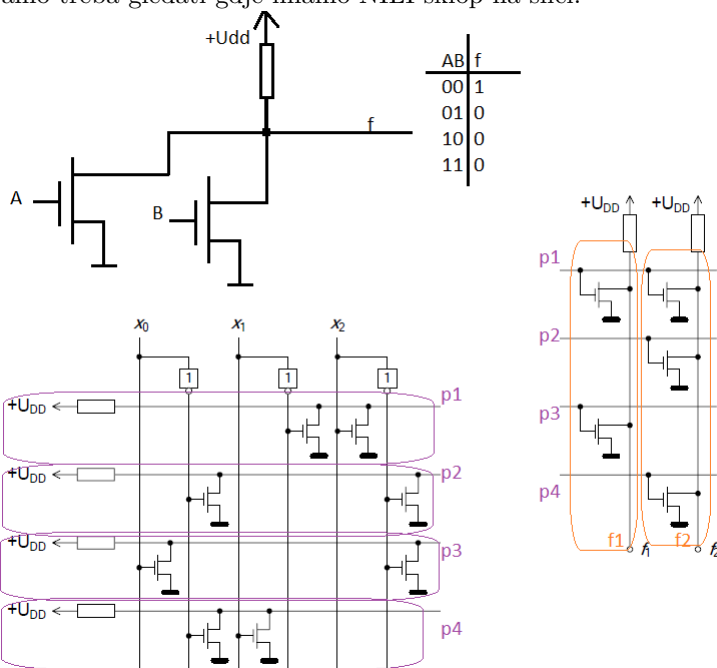
Iz tablice možemo vidjeti $y_1 = \sum m(2, 3) = x_0$ i $y_0 = \sum m(0, 2) = \bar{x}_0$.

Općenito je korisna formula za pronalazak B-1 komplemeneta u sustavu s bazom B (Z je znamenka): $Z + \bar{Z}^{B-1} = B - 1$

23. PLA strukturom u tehnologiji MOSFET ostvarene su funkcije f_1 i f_2 (vidi sliku). O kojim se funkcijama radi?



Samo treba gledati gdje imamo NILI sklop na slici.



Dvije razine NI-NI se mogu pretvoriti u I-ILI (suma produkata) tako da otkotiramo invertore na ulaze iduæeg NI, a NILI-NILI u ILI-I (produkt suma).

Mi sa slike itamo:

$$p_1 = x_2 N I L I \bar{x}_1$$

$$p_2 = \bar{x}_2 N I L I \bar{x}_0$$

$$p_3 = x_0 N I L I \bar{x}_2$$

$$p_4 = x_1 N I L I \bar{x}_0$$

$$f_1 = p_1 N I L I p_3 = x_2 * x_0 + \bar{x}_2 * \bar{x}_1$$

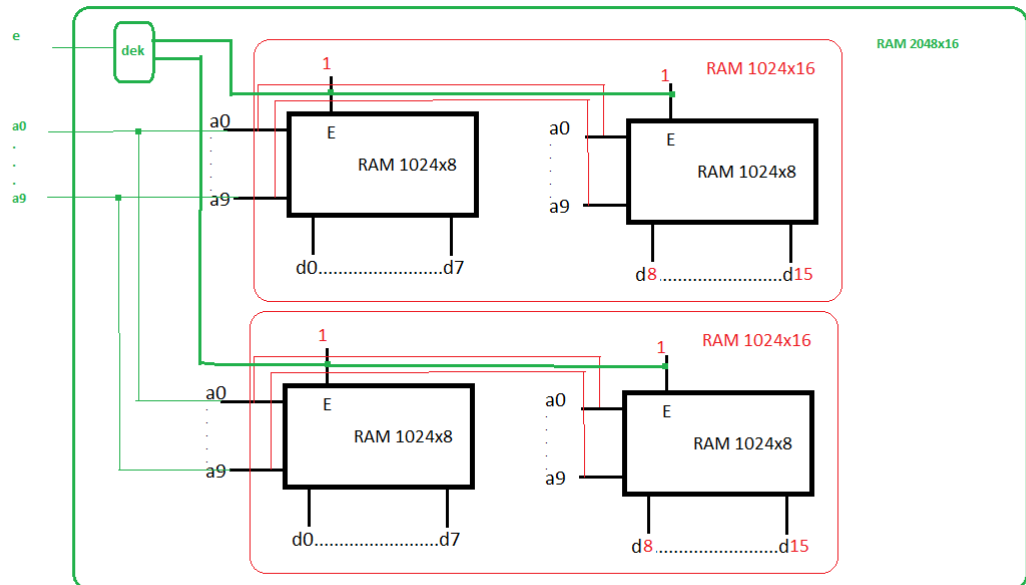
$$f_2 = p_1 N I L I p_2 N I L I p_4 = x_2 * \bar{x}_0 + \bar{x}_1 * \bar{x}_0$$

- 24 Na raspolaganju su čipovi RAM-a kapaciteta 1024x8 bita. Potrebno je izgraditi RAM kapaciteta $2^{14} \times 32$ bita. Koliko adresnih ulaza treba imati dekodera koji upravlja ulazima za omogućavanje korištenih manjih čipova RAM-a?

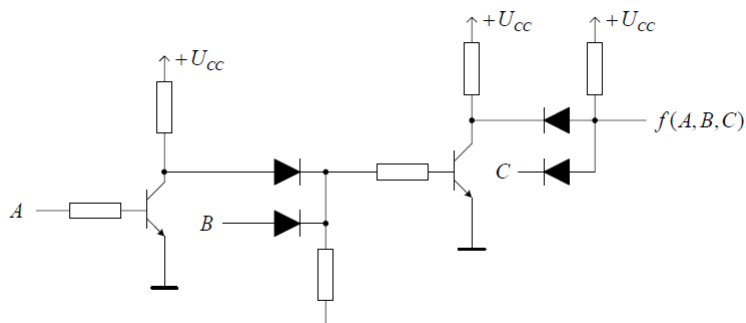
Za $2^{10} \times 2^3$ RAM imamo 8 podatkovnih izlaza. Trebamo ostvariti $2^{14} \times 2^5$ RAM. Ako želimo ostvariti $2^{10} \times 2^4$ RAM imamo dva $2^{10} \times 2^3$ RAM u paraleli, te dijele adresne ulaze. Što ako želimo $2^{11} \times 2^4$ RAM? Tada imamo dva retka $2^{10} \times 2^4$ RAM (samo dodamo još jedan redak) sa istim adresnim ulazima. Povezujemo žicom kao u zadatku 17. pa treba paziti da ne bi slučajno na tu žicu došla dva podatka iz raznih RAM-ova u drugim retcima. Da se to osigura, igramo se enable ulazom (omogućavanje). Za 1024x16 RAM imamo 10 adresnih bitova, 10 "normalnih" i 1 za dekodera koji upravlja s enable ulazima.

Za ostvarivanje $2^{10} \times 2^3 \rightarrow 2^{14} \times 2^5$ imamo 4 čipa u retku, 16 redaka. Zbog 16 redaka nam treba 16 izlaza iz dekodera za enable pa je to dek 4/16, tj. imati će 4 adresna ulaza.

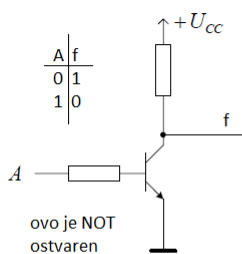
Evo i slikica da bude jasnije:



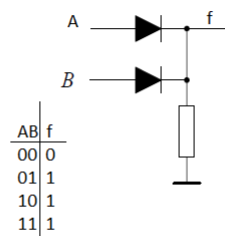
25. Koju funkciju u pozitivnoj logici obavlja sklop prikazan na slici? Prikažite tu funkciju kao sumu minterma.



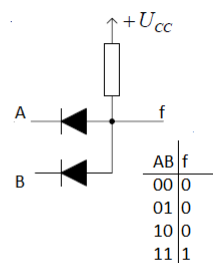
Analizirajmo dijelove:



ovo je NOT
ostvaren
bipolarnim
tranzistorom - radi kao
n-kanalni MOSFET

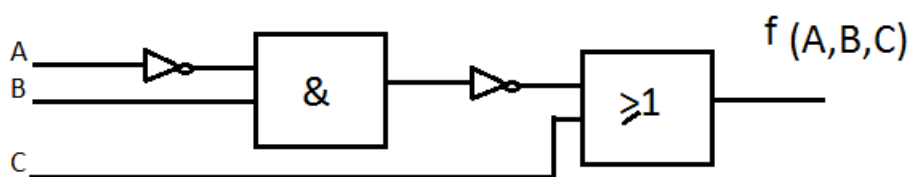


ovo je logički ILI sklop



ovo je logički I sklop

Sad konstruiramo funkciju:



Direktno vidimo da je funkcija $f = \neg(\neg A + B) * C = A * \neg B * C = m_5$

Preporuča se pogledati e-auditorne i Čupićevu zbirku-dosta zadataka ima sličan princip. Rješen jedan prošli ZI nije naučen predmet!

Ako vidite kakve sintaktičke pogreške javite da ispravim.

E-mail address: amilinov@dominis.phy.hr