

DIGLOG MI2

-postupci (B grupa)

Wolfman

Nadam se da će ovo barem nekom pomoći. Ako pronađete kakvu grešku ili ako vam nešto nije jasno slobodno se javite na PM ili mail (bnovoselnik@gmail.com) s pitanjima.

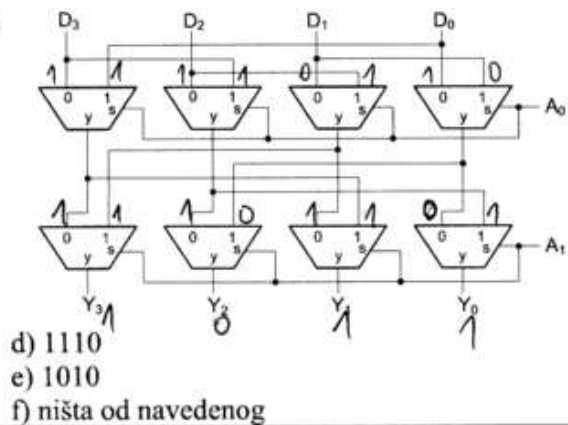
Sretno u 3. ciklusu :D

Wolfman

P.S. Ovdje je riješena grupa B, no princip je isti za sve grupe!

Wolfman

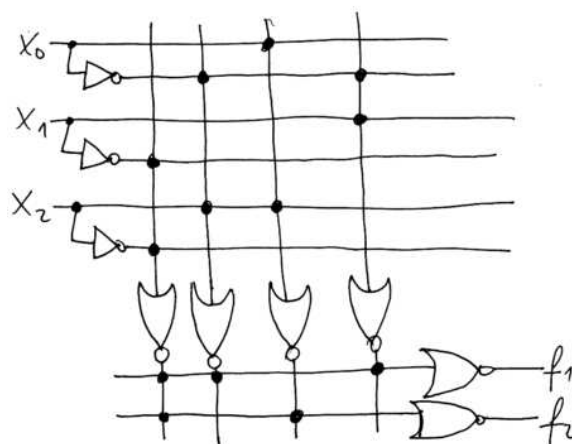
1. Što će biti na izlazima $Y_3Y_2Y_1Y_0$ sklopa sa slike, ako se na ulaze dovede $D_3D_2D_1D_0=1101$, $A_1A_0=11$?



Vjerujem da svi na slici prepoznaju osnovni sklop – **multipleksor** 2/1. Ponovimo koja je njegova funkcija. Multipleksor na svoj izlaz propušta **jedan** od podataka na podatkovnim ulazima, ovisno o **seleksijskom** ulazu. Ako pogledate oznake na muxovima u zadatku vidjet ćete oznaku s. To je seleksijski ulaz. Iz slike se vidi da A_0 selektira podatkovne ulaze u gornjem redu muxova dok A_1 radi to isto samo u donjem redu. Podatkovni ulazi na muxovima su obilježeni sa 0 i 1, što znači da se propušta podatak na ulazu 1 kada je seleksijski ulaz 1, tj. podatak na ulazu 0 kada je seleksijski ulaz 0.

Ovaj zadatak je u biti jako lagan (ako znamo ove osnovne stvari o multipleksorima i ako smo dovoljno koncentrirani :D). Samo treba pratiti gdje je što spojeno. U zad je zadano $D_3D_2D_1D_0=1101$. Upišimo to na sliku (to su ulazi označeni nulom u gornjem redu muxova). Sada pratimo žice i upišimo što se dovodi na ulaze označene jedinicom ($\Rightarrow 1110$). Također je zadano da su oba sel. ulaza 1, što znači da će se propustiti ulazi označeni jedinicom. Pa „propustimo“ ih (upisujemo 1110 na ulaze označene nulom u donjem redu muxova – jer su izlazi iz gornjeg reda spojeni upravo na te ulaze u donjem redu). Sada opet pratimo žice i upišimo što imamo na ulazima označenima jedinicom ($\Rightarrow 1011$). Budući da je sel ulaz opet 1, propuštamo ono što je na ulazima označenima jedinicom pa je rješenje c) 1011.

2.



Sve što ovdje treba znati je da je PLA struktura ostvarena MOSFET-ima zapravo **NILI-NILI** matrica.

Na slici je to i nacrtano tako radi lakšeg snalaženja. Znači svagdje gdje je na crtežu zadatka nacrtan mosfet ide moja točka (znači tu je spoj). Sada samo treba očitati funkciju.

Pogledajmo prvu okomitu liniju. Tu imamo točke na not x_1 i not x_2 . To se sve vodi na NILI sklop znači rezultat je $\text{NILI}(x_1', x_2')$ što je jednako (upotrijebimo **de Morgana**) x_1 and x_2 . Na isti način dođemo do „rezultata“ ostalih okomitih linija.

2. linija = x_0 and x_2'

3. linija = x_0' and x_2'

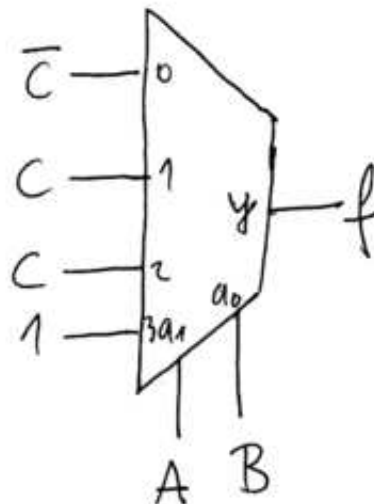
4. linija = x_0 and x_1'

Da bi dobili f_1 moramo rezultate 1., 2., i 4. (primjetite točkice) linije dovesti na NILI sklop. Kada to opet raspišemo po de Morgana dobijemo $f_1 = (x_1' + x_2')(x_0' + x_2)(x_0' + x_1)$.

Primjetimo sada da smo to mogli i izravno pročitati iz matrice. Samo za svaku okomitu liniju čitamo varijable na kojima je točka i upisujemo ih kao sume. Na kraju sve sume pomnožimo i dobili smo traženu funkciju. Dakle iz prve linije čitamo $x_1' + x_2'$, iz druge $x_0' + x_2$ i iz 4. (primjetite da nam za f_1 ne treba 3. linija) $x_0' + x_1$, pa je $f_1 = (x_1' + x_2')(x_0' + x_2)(x_0' + x_1)$. Ovo vrijedi za sve NILI-NILI matrice. Vrlo je korisno jer vam drastično skraćuje vrijeme potrebno za rješavanje ovakvih zadataka, jer kao što je netko jednom rekao – vrijeme je zadatak :D Nakon što u f_1 sve izmnožimo i minimiziramo koristeći osnovne aksiome Boolove algebre dobijemo da je $f_1 = x_2'x_0' + x_1'x_0'$. Vidimo da samo a) odgovor ima takvu f_1 pa odmah zaokružujemo a) i krećemo na slijedeći zadatak (opet se vodimo pravilom vrijeme je zadatak :D).

3.

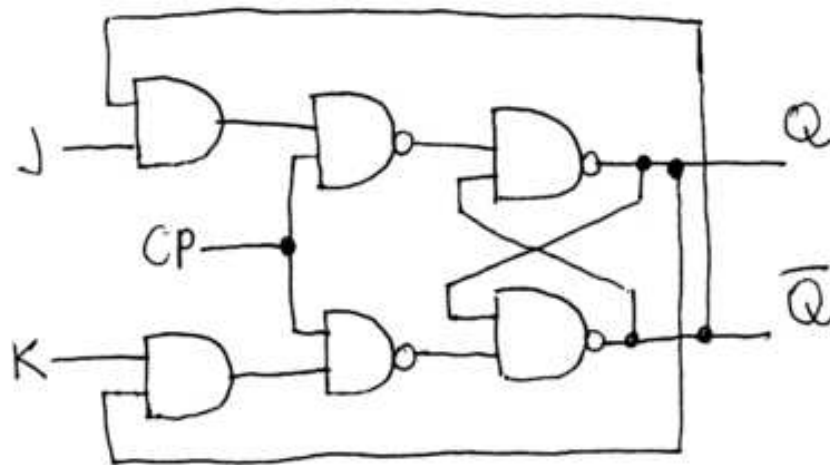
A	B	C	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Ako ne adresne ulaze dovodimo A i B onda će se rezidualne funkcije (one koje dovodimo na podatkovne ulaze D0 – D3) sastojati od najviše jedne varijable (i to je varijabla C). Ovaka se zadatak najlakše riješi tablicom. Znači prvo napišete tablicu kombinacija. Zatim jednom okomitom linijom odvojite varijable koje su na selekcijskim ulazima (u ovom slučaju su to A i B). Sada vidite da su u prva dva reda A i B 00, zatim u slijedeća dva 01 itd do 11. Kada je recimo $AB=00$ funkcija ovisi samo o varijabli C. Ako je $C=0$ fja je 1, a ako je $C=1$ fja je 0. Drugim riječima kada je $AB=00$, fja je NOT C. Slično

razmatranje slijedi za ostale kombinacije (vidi se na slici). I to je upravo ono što trebamo dovesti na podatkovne ulaze muxa. Još jednom se prisjetimo. Kada je $AB=00$, propustit će se podatak na ulazu označenom s 0. Mi želimo da se propusti upravo vrijednost funkcije koja ovisi još i o varijabli C. Iz tablice očitamo čemu je fja jednaka kada je $AB=00$. Jednaka je upravo NOT C. Vid vraga ! Znači upravo to ćemo spojiti na ulaz D0. Analogno zaključujemo za ostale ulaze.

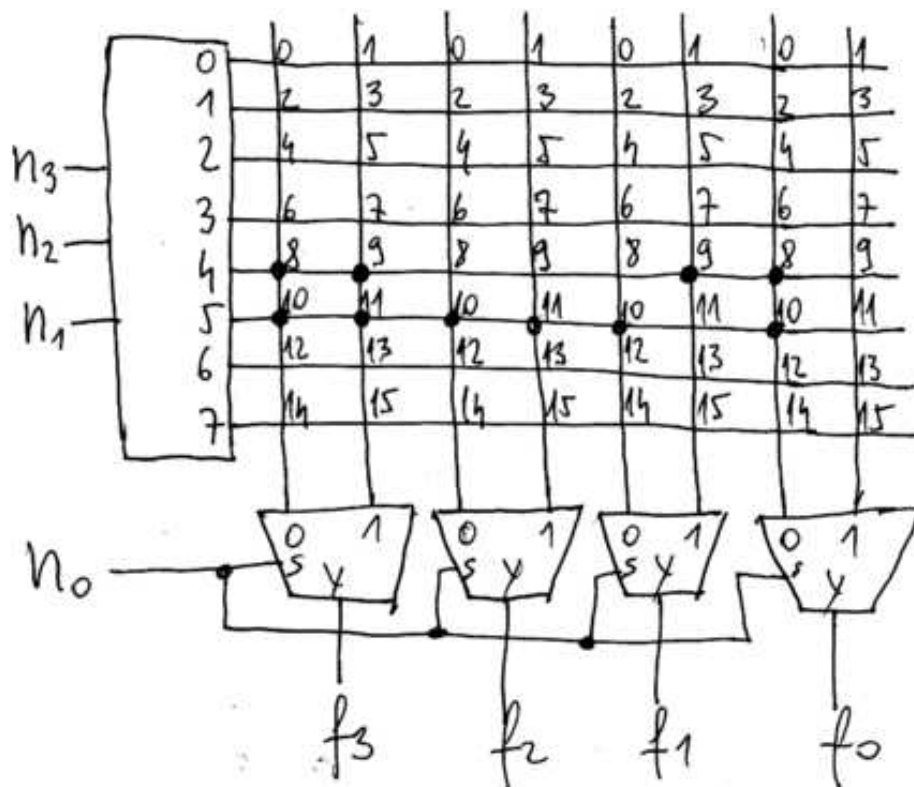
4.



Na slici je prikazana shema sklopa. Ako ovo znate to vam je pola riješenom zadatka. Ja sam krenuo ovom logikom (neka me netko ispravi ako sam pogriješio pa slučajno došao do točnog odgovora :D):

Ako je kašnjenje svakom sklopa 10 ns, nema šanse da se u 125 ns promjeni stanje pa u 125 ns je još (0,1) – znači eliminiramo sve odgovore osim c) i d). U 130. ns mijenja se stanje u (1,1) (pratite redom promjene na sklopovima u koracima po 10 ns..to mi se ne da sada pisati no sa shemom mislim da uopće nije teško doći do toga) i tako ostaje sve do 140. ns kada se mijenja u (1,0), što znači da je u 135. još uvijek (1,1) pa smo eliminirali i c). Dakle jedini točan ostaje d) pa zato njega i zaokružujemo i brzinski prelazimo na 5. zad. :D

5.



E ovaj zadatak treba prvo par puta pročitati da bi se shvatilo što pjesnik hoće reći. Dakle, kaže da treba ostvariti funkciju $f(n) = n \text{ EXILI } n^{\wedge}$, gdje je n^{\wedge} broj n zarotiran za 1 mjesto lijevo.

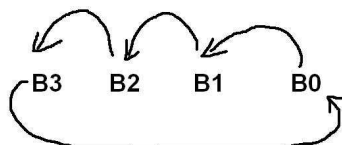
Prvo da vidimo što to znači.

Zarotirati binarni broj za jedno mjesto lijevo u ovom slučaju očito znači „posmaknuti“ ga i to kružno (za jedno mjesto lijevo naravno :D). Uzmimo primjer:

$n = 1001$

$n^{\wedge} = 0011$

Drugim riječima napravimo ovo (tj slikom :D, no slika govori 1000 riječi pa...):



U drugim grupama je zadana drugačija vrsta posmaka no to neću posebno pojašnjavati (imate to na slajdovima).

E sad, za svaki broj od 0 do 15 (tj za svaku kombinaciju u tablici) treba napraviti ovaj posmak i onda još XOR-ati ta dva broja i to je rezultat funkcije. Funkciju trebamo ostvariti ispisnom memorijom 8x8 (na slici). Idemo prvo objasniti sliku.

Kao što znate, memorija ima jedan dekoderski dio. To je dekodler na slici. Ulazi su mu n_3 - n_1 . Oni nam određuju memorijsku lokaciju. Memorijska lokacija je figurativno rečeno svaka vodoravna žica na crtežu. Znači ima 8 memorijskih lokacija i svaka je 8-bitna (tako je zadano – 8x8). E sad za selekciju se koriste muxovi kojima upravlja četvrta varijabla n_0 . Pogledajmo prvo tablicu:

N	n_3	n_2	n_1	n_0	n^{\wedge}	f
0	0	0	0	0	0000	0000
1	0	0	0	1	0010	0011
2	0	0	1	0	0100	0110
3	0	0	1	1	0110	0101
4	0	1	0	0	1000	1100
5	0	1	0	1	1010	1111
6	0	1	1	0	1100	1010
7	0	1	1	1	1110	1001
8	1	0	0	0	0001	1001
9	1	0	0	1	0011	1010
10	1	0	1	0	0101	1111
11	1	0	1	1	0111	1100
12	1	1	0	0	1001	0101
13	1	1	0	1	1011	0110
14	1	1	1	0	1101	0011
15	1	1	1	1	1111	0000

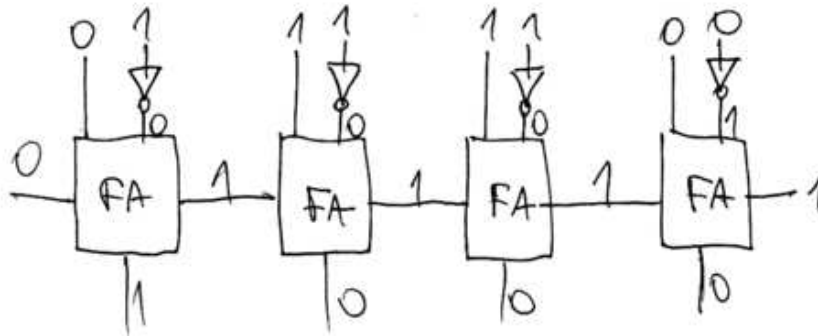
Kada je $n_3n_2n_1=000$ na dekoderu se aktivira nulta vodoravna žica. Sada postoje dva slučaja – $n_0=0$ i $n_0=1$. Kada je $n_0=0$ na muxove se propušta ono što je na sjecištima trenutne vodoravne žice (nulte) i okomitih žica koje su spojene na muxove ulaze označene nulama. I na tim sjecištima ćemo onda upisati vrijednost funkcije za $n_3n_2n_1n_0=0000$. Za drugi slučaj, kada je $n_0=1$, van ide ono što je na sjecištima sa okomitim žicama koje vode na ulaze označene jedinicom, pa ćemo tu upisati vrijednost funkcije za kombinaciju $n_3n_2n_1n_0=0001$. Kao što vidite to sam sve obilježio na crtežu. Četiri nule u prvom redu znače da se na ta mjesta upisuju vrijednosti bitova funkcije (jer se fja sastoji od 4 bita). Istom logikom su označena i ostala mjesta. Ako želimo upisati 1 staviti ćemo točkicu, a ako želimo nulu ostavit ćemo prazno. Sada kada je to jasno (nadam se da je :D), krenimo dalje.

Ako ste dobro pročitali zadatak primjetili ste da ne morate izračunavati sve ove vrijednosti. U rješenjima su ponuđene vrijednosti memorijskih lokacija od 4 do 7. Uzmimo lokaciju 4. To je naša žica broj 4 na crtežu. Na njoj su upisane vrijednosti funkcije za broj 8 i broj 9. Ako pogledamo u tablicu to su redom: $f(8)=1001$, te $f(9)=1010$. To unosimo na crtež tako da na mjesta gdje su osmice redom upisujemo 1 0 0 1, a za devetke 1 0 1 0. Na kraju dobijemo slijed bitova 11000110. Ta vrijednost je upisana na memorijsku lokaciju broj 4. Zadatak kaže da su sadržaji memorijskih lokacija u heksadekadskom zapisu pa ćemo to prebaciti u heksa zapis.

$$11000110_2 = C6_{16}$$

Vidimo da jedino odgovor a) ima ponuđeno C6 pa je to točan odgovor. Možemo još reda radi provjeriti i za lokaciju 5. Istim postupkom dobivamo slijed 11111010 što je u heksa zapisu FA. Sad smo još sigurniji u točnost svog rješenja pa sretni možemo nastaviti dalje ☺

6.



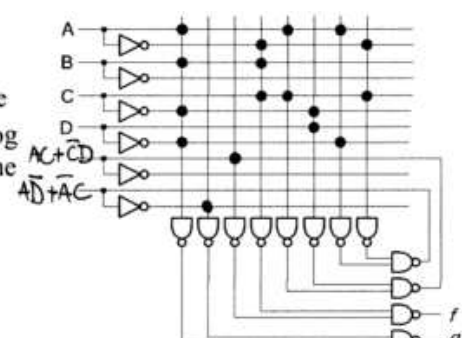
Na slici je shema sklopa. Zadane su vrijednosti $a=0110$ i $b=1110$. Trebamo obaviti operaciju $a-b$ pomoću sklopa za zbrajanje. Kako ćemo to? Sjetimo se da se oduzimanje može svesti na zbrajanje s B komplementom. Sjetimo se da se B-1 komplement dobiva jednostavnim invertiranjem znamenaka (u bazi 2), a B komplement dodavanjem jedinice B-1 komplementu. Znači $a-b$ možemo zapisati kao $a+b^*$ gdje je b^* B-komplement broja b.

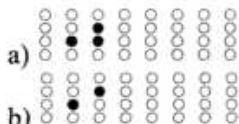

Čime mi raspolažemo? Sa 4 invertora. Oni će nam baš lijepo poslužiti kao invertori bitova broja b, tj. pomoću njih ćemo dobiti B-1 komplement. No što sada? Nama treba B komplement ☹ Fear not! Sjetimo se da sklop FA (full adder, iliti po naški sklop za potpuno zbrajanje) kao jedan od ulaza ima i prijenos sa nižeg mjesta. Kada smo ga koristili za zbrajanje na prvi FA sklop smo dovodili 0 kao minus prvi prijenos. Sada ćemo jednostavno dovesti 1 kao taj prijenos jer će nam se ta jedinica onda zbrojiti sa B-1 komplementom i dobit ćemo upravo B komplement, a to je ono što nam treba da bi pravilno oduzeli ova dva broja. Dalje je sve jasno (nadam se). Na shemi je pisano koliki je pojedini ostatak tako da je rezultat na kraju $c4c3c2c1=0111$

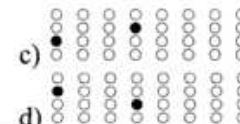

7. Ovaj zadatak neću posebno pojašnjavati. Opet se radi o NILI-NILI matrici. Samo treba očitati i minimizirati funkcije na način objašnjen u 2. zadatku.

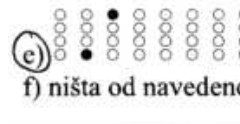
8.

8. Sklopom PAL prikazanim na slici potrebno je ostvariti funkcije: $f = \overline{A}BC + AC + \overline{C}D$ i $g = A\overline{B}\overline{C}\overline{D} + (\overline{A} + D)(A + \overline{C})$. Dio programiranja već je ostvaren. Kako treba programirati posljednja 4 retka prvog polja kako bi prikazana struktura doista ostvarivala zadane funkcije?



a)  b) 

c)  d) 

e)  f) ništa od navedenog

Opet vrlo slična stvar. Sada se radi o NI-NI polju. Vidimo da se dva člana vraćaju nazad u prvu NI ravninu. Odredimo prvo njih.

Opet gledate gdje su točke i čitate. Svaka okomita linija je produkt, a ukupna funkcija na kraju je suma tih produkata. Prva fja koja se vraća u NI ravninu je $AC + C'D$. To smo izravno pročitati na napisani način. Druga je $AD' + A'C$.

Naše funkcije f i g su zadane. Moramo isprogramirati zadnja 4 reda da bi na izlazima dobili zadane fje.

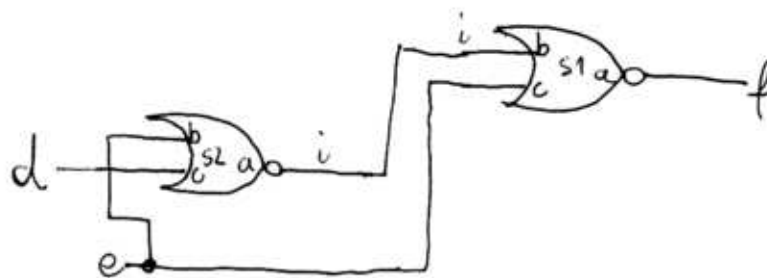
E sada, ovo možete na više načina, no možda je najlakše eliminacijom. Isprobate i brzo pročitate i ako štima super, ako ne idemo dalje xD. Uzmimo recimo e) Unesimo točke i pročitajmo.

$$f = A'BC + AC + C'D$$

$$g = ABC'D' + (AD' + A'C)' = ABC'D' + (A' + D)(A + C')$$

Tko bi rekao! Pogodili smo iz prve! Jeej! Znači točno je e).

9.

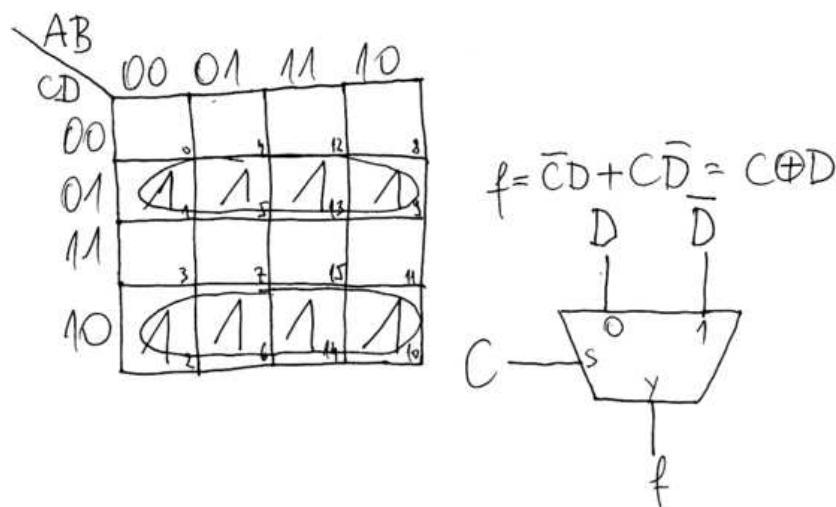


Ovo je isto jedan od lakših. Samo trebate iz strukturnog opisa nacrtati shemu (ili možete i napamet no ovako je valjda lakše ne zeznut :D) i onda pročitati funkciju. Shema je na slici. Nacrtana je na

slijedeći način. U strukturnom opisu piše da su portovi instance s1 spojeni na slijedeći način: na ulaz b spojena je žica i, na ulaz c signal e i na izlaz a signal f (tj. naša funkcija). To je zadano eksplicitno. Druga instanca je zadana implicitno. Tu je važan redoslijed portova (kojim redom su deklarirani). U ovom slučaju prvo ide a pa onda b i na kraju c (pogledati sučelje sklopa nor2). Stoga su redom na a spojena žica i, na ulaz b signal e i na ulaz c signal d. Iz sheme slijedi da je

$$f = \text{NILI}(e, \text{NILI}(e, d)) = de' = d \text{ and } (\text{not } e);$$

10.



I ovo je jedan od zadataka koji bi bilo dobro pročitati do kraja (pogotov razno razne napomene :D). Recimo da smo mi iz prve pročitali napomenu i odlučili pogledati minimalni zapis fje.

Vidi! Pa fja uopće ne ovisi o varijablama A i B. Svodi se na fju C XOR D što se lako da ostvariti jednim jedinim muxom 2/1 (prikazano na slici)!

11. Ako rezidualne funkcije smiju imati najviše jednu varijablu (a imamo ukupno 7 varijabli) onda moramo izgraditi takvo multipleksorsko stablo koje će imati 6 selekcijskih ulaza (da..jer je $7-6=1$:D). Tako nam uistinu ostaje samo jedna varijabla koju možemo dovoditi na podatkovne ulaze. Ako imamo 6 sel ulaza, tada ima $2^6=64$ podatkovnih ulaza. Ako imamo muxove 4/1 to znači da ćemo u prvo razini stalba imati $64/4=16$ muxova. Tih 16 muxova će imati 16 izlaza, a izlaze opet skupljamo muxovima 4/1 pa ćemo u drugoj razini imati $16/4=4$ muxa. I konačno njihove izlaze može pokupiti samo jednim muxom i to je vrh stabla. Dakle, ukupno imamo $16+4+1=21$ mux pa je odgovor a).

12.

Kada je na ulazima dekodera $AB=00$, izlaz 0 je postavljen u jedinicu, a svi ostali u 0. To znači da su podatkovni ulazi muxa **1000** (paziti i na onaj NULI sklop), pa na prva četiri mjesta u tablici upisujemo upravo te vrijednosti (primjetimo da se dok je $AB=00$ CD mijenja od 00 do 11, tj propušta redom ulaz 0 do ulaza 3). Za $AB=01$ izlaz 1 je postavljen u jedan a svi ostali u nula, pa su ulazi muxa redom **0000**. Za $AB=10$ imamo **0110**, te za $AB=11$ vrijednosti su **0101**.

A	B	C	D	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Minimizirate metodom po izboru i dobijete $f=A'B'C'D' + AB'CD' + AC'D + ABD$

13. Označimo sa f_1 funkciju između A i B, a sa f_2 funkciju između C i D.

Te fje su označene preglednim tablicama, koje efektivno znače vrijednost funkcije za pojedinu kombinaciju. Tako npr vrijedi

A	B	f1
0	0	0
0	1	1
1	0	1
1	1	0

Pa je $f_1=A \text{ XOR } B$.

C	D	f2
0	0	1
0	1	0
1	0	0
1	1	0

Iz ovoga slijedi da je $f_2=C'D'$

Konačno vrijedi:

f1	f2	f
0	0	0
0	1	0
1	0	0
1	1	1

Drugim riječima: $f=f1$ and $f2$.

$f = (A \text{ XOR } B) \text{ and } (\text{not } C \text{ and } D) = (\text{de Morgan}) = (A \text{ xor } B) \text{ and not}(C \text{ or } D)$

14.

Prvo pišemo tablicu sklopa koji želimo dobiti:

Qn	X	Y	Qn+1
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Napišimo tablicu (uzbudnu) sklopa kojim raspolažemo:

Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

Sada proširimo početnu tablicu sa T (tamo gdje su Qn i Qn+1 različit stavljam 1, 0 u suprotnom...to se vidi iz uzbudne tablice):

Qn	X	Y	Qn+1	T
0	0	0	1	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Minimizacijom slijedi točan odgovor pod d).

15.

Kodirajmo brojeve po uputama. Drugi dio tablice (k3k2k1k0) su B-1 komplementi znamenaka. Budući da je baza 10 to je 9 komplement. 9 komplement znamenke b se dobije ovako $b^* = 9 - b$. Znači komplement znamenke 9 je 0, a znamenke 0 je 9, znamenke 8 je 1, a jedinice je 8 itd.

n	B3	B2	B1	B0	K3	K2	K1	K0
0	0	0	0	1	1	0	1	1
1	0	0	1	0	1	0	1	0
2	0	0	1	1	1	0	0	1
3	0	1	0	0	1	0	0	0
4	0	1	0	1	0	1	1	1
5	0	1	1	1	0	1	0	1
6	1	0	0	0	0	1	0	0
7	1	0	0	1	0	0	1	1
8	1	0	1	0	0	0	1	0
9	1	0	1	1	0	0	0	1
10	X	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X
12	X	X	X	X	X	X	X	X
13	X	X	X	X	X	X	X	X
14	X	X	X	X	X	X	X	X
15	X	X	X	X	X	X	X	X

Sada treba minimizirati fju k2, no treba paziti da su mintermi izmješani.

$$K2 = m(5,7,8) + d(0,6,12,13,14,15).$$

X		X	1
	1	X	
	1	X	
	X	X	

Kada minimizirate odaberite onaj minimizirani oblik koji je ponuđen u rješenjima i to je to. Da bi dobili rješenje b) trebate zaokružiti ova 4 crvena u sredini i plava dva u gornjem redu. I to je to!