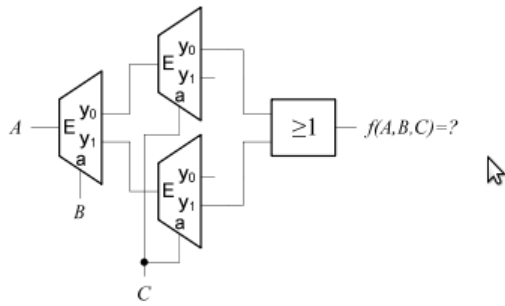


# Čupčevi zadaci !

2.Mi grupa D, 2010-2011, nepismen sam i nije me briga , ako nesto ne mozete shvatit javite se mitru

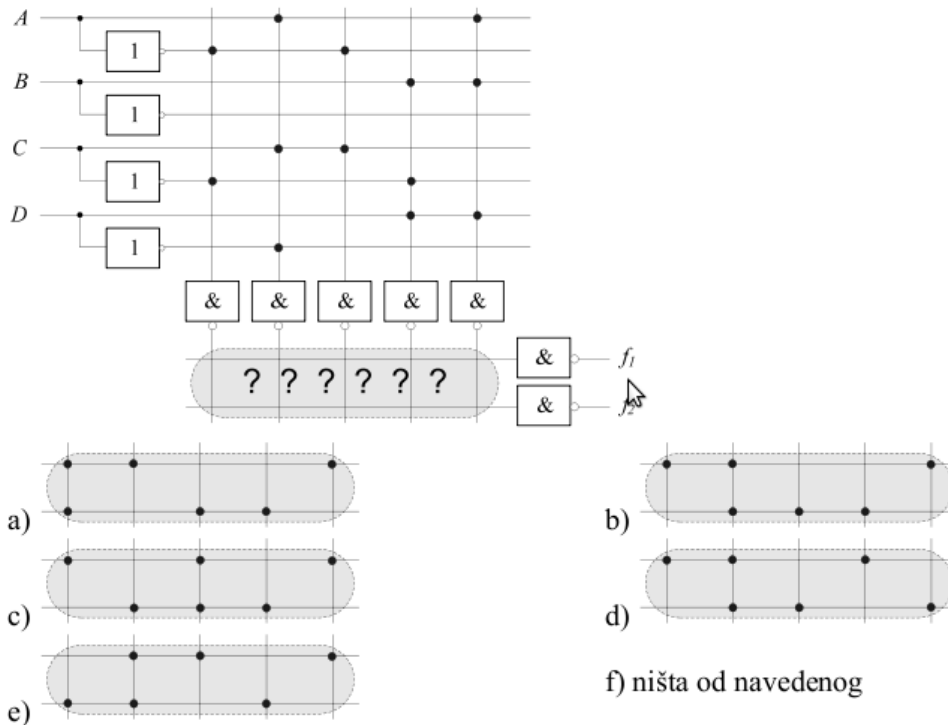
1. Sklop s tri dekodera i jednim logičkim sklopom prikazan je na slici. Kako glasi zapis njegova izlaza  $f(A,B,C)$ ?



- a)  $A \cdot \overline{B \oplus C}$                       c)  $A \cdot (B \oplus C)$                       e)  $A \cdot (B + \overline{C})$   
b)  $A \oplus (B \cdot C)$                       d)  $\overline{A} + B \cdot C$                       f) ništa od navedenoga

1. Imamo 3 multipleksora i jedan sklop ili . Vidimo da je A spojen na Enable a kod multipleksora to znaci da ce se A mnoziti sa B i sad gledamo: Imamo  $A \times B$  i  $A \times \text{not } B$  (po  $y_0$  i  $y_1$  vidimo kad je B 0 a kad je 1), sto nam daje A and B i A and not B. I sad dolazimo na druge multipleksore a ovo dvoje sto smo dobili nam je enable . Imamo A and B i vidimo da to ide na donji multipleksor koji propusta C i dobivamo A and B and C . Gledamo gornji , tamo imamo A and not B i taj propusta not C pa imamo , A and not B and not C, I to sve zbrojimo zbog sklopa ili i dobijemo  $f=ABC$  or A and not B and not C. Vidimo da to nema nigdje ponudjeno pa moramo traziti , ako trazimo po redu srecom naletimo da je A točno i mozemo dalje :D.A!

2. PLA sklop tipa NI-NI treba programirati tako da ostvaruje funkcije  $f_1 = \overline{A}\overline{C} + AC\overline{D} + ABD$  i  $f_2 = \overline{A}C + AC\overline{D} + B\overline{C}D$ . PLA sklop je prikazan slikom i već je djelomično programiran. Kako treba programirati njegovu drugu ravninu (zasivljeno područje)?



2. Imamo NI-NI i tu znamo (valjda znamo :P) da kad imamo NI-NI samo stavimo minterme i to je to , a za slucaj da imamo NILI-NILI trebamo koristiti maxterme.

I sad gledamo , imamo NI-NI i samo trebamo minterme pogledat,  
 Gledamo za  $f_1$ =vidimo da je – not A and not C na prvoj tocki(prva okomita linija znaci da nju koristimo ,  
 idemo dalje - A and C and not D vidimo da je na na drugoj okomitoj liniji znaci i tu tocku stavljamo i idemo dalje  
 zadnji clan funkcije trazimo- A and B and D i vidimo da je na zadnjoj okomitoj liniji i tu stavljamo tocku . Tocke su nam na 1. 2. 5. mjestu( a ili b je točno)  
 Gledamo za  $f_2$  (gledat cemo samo drugu tocku i odmah vidjet koje je točno)

na drugoj okomitoj liniji imamo A and C and not D(sto imamo i ponudjeno) a vidimo da samo ponudjeno pod B ima tocku dva i zaokruzujemo B.B!

3.	<p>Sklopove PAL označavamo oznakom <math>m \times k \times n \times n</math>, pri čemu <math>m</math> predstavlja broj ulaza, <math>n</math> broj izlaza a <math>k</math> broj ulaza u pojedini izlazni sklop NILI. Takav sklop želimo programirati tako da na svojem ulazu dobije tri bita: <math>a</math>, <math>b</math> i <math>c</math>. Sklop na svojim izlazima treba generirati sumu, prijenos, razliku i posudbu (dakle, PAL treba ostvariti funkciju potpunog zbrajala i potpunog oduzimala). PAL koji imamo na raspolaganju izveden je pomoću dvije razine NILI sklopova. Ako implementacija svih funkcija mora biti dvorazinska, koje su minimalne dimenzije sklopa PAL koji nam je potreban?</p>		
	a) $3 \times 12 \times 3$	c) $3 \times 6 \times 3$	e) $3 \times 3 \times 3$
	b) $3 \times 9 \times 3$	d) $3 \times 15 \times 3$	f) ništa od navedenoga

3. Nisam siguran za ovaj ali sam ga je ovako gledao . 3 ulaza 3 izlaza i sad jos trebamo k , posto trebamo sumu, prijenos , razliku i posudbu ja sam uzeo da je k 4 i dobijem  $3 \times 12 \times 3$  sto je A i točno :D.A!

4.	<p>Na 4-bitno binarno zbrajalo sa serijskim prijenosom izvedenim uporabom 4 potpuna zbrajala u trenutku <math>t = 0</math> ns doveden je podatak <math>a=0000</math>, <math>b=0000</math> te <math>c_{in}=0</math>. U trenutku <math>t = 200</math> ns dovode se podatci <math>a=1001</math>, <math>b=1100</math> te <math>c_{in}=1</math>. Izlaz <math>c_{out}</math> potpunog zbrajala kasni 20 ns a izlaz bita rezultata kasni 30 ns. Od kojeg trenutka će i rezultat zbrajanja i konačni prijenos biti ispravni za <b>podatak i ulazni prijenos</b> doveden u <math>t=200</math> ns?</p>		
	a) od $t=320$ ns	c) od 280 ns	e) od 230 ns
	b) od $t=250$ ns	d) od 400 ns	f) ništa od navedenoga

4. Morat ćemo zbrojiti 1001 i 1100 i dodatak 1 i sad kad napisemo

1+1 0+1 0+0 1+0+1 (prvi članovi , drugi članovi,...,zadnji članovi+cin)  
dobivamo da su prva 2 zbrajala ok jer se na njima kasnije neće ništa promijeniti  
samo gledamo zadnja 2 . Vidimo da cout kasni 20 ns a rezultat 30 ns i sad zbroji  
se 0+0, 1+0+1 i dobijemo 0 i 0, na 3 prijenos 0 a na 4 prijenos 1 (cout) on kasni 20  
ns i ide na 3 zbrajalo koje kasni 30ns, i racuna se 0+1 sto nam daje 1 i nakon  
20+30 ns dobivamo ispravan rezultat sto nam daje 200+50ns i odgovor pod B sto je  
točan :D.B!

5. Ne znam :D

6.	Kvartarne znamenke kodiraju se pomoću 3 bita, pri čemu je znamenka $i$ kodirana kao binarno zapisana vrijednost $2i+1$ . Potrebno je projektirati sklop koji na svojem ulazu $a_2a_1a_0$ prima kod kvartarne znamenke a na svojem izlazu $r_2r_1r_0$ generira njezin 3-komplement (u istom kodu). Za ulaz koji ne predstavlja valjani kod znamenke izlaz nije bitan. Odredite minimalni zapis funkcije $r_1$ .		
a)	$a_2 + \bar{a}_1a_0$	c)	$a_2a_1 + \bar{a}_0$
b)	$\bar{a}_2 + \bar{a}_1a_0 + a_1\bar{a}_0$	d)	$\bar{a}_2$
		e)	$\bar{a}_1$
		f)	ništa od navedenoga

6. Imamo kvartarni sustav i brojeve 0,1,2,3 . Svaka se kodira kao  $2i+1$  .

0 kao  $0 \times 2 + 1 = 1$  ,  $1 \times 2 + 1 = 3$  ,  $2 \times 2 + 1 = 5$  ,  $3 \times 2 + 1 = 7$  .

Sto nam govori da je 0 kodirano kao binarna 1 (001), 1(binarna 3 011), 2(binarna 5 101), 3(binarna 7 111). Komplementi su : od 0 je 3, 1 je 2, 2 je 1, 3 je 0.

Sad pisemo tablicu :

pisemo tablicu za sve kombinacije i u  $r_2r_1r_0$  pisemo komplemente(brojeve koje ne koristimo stavljamo kao not care – sto ce na koristit pri minimalizaciji)

000 ne koristimo pa je to xxx, 001 koristimo sto je 0 a komplement je 3 pa pisemo kod za 3, 010 ne koristimo pa je to xxx, 011 koristimo kao 1 pa tamo pisemo kod za 2 a to je binarna 5, 100 ne koristimo pa je to xxx, 101 koristimo kao 2 i tamo pisemo kod za jedinicu (binarna 3), 110 je xxx , 111 je 3 a komplement je 0 sto nam je 001

$a_2a_1a_0$  |  $r_2r_1r_0$

0 0 0	x x x
0 0 1	1 1 1
0 1 0	x x x
0 1 1	1 0 1
1 0 0	x x x
1 0 1	0 1 1
1 1 0	x x x
1 1 1	0 0 1

Kad imamo ovu tablicu minimiziramo  $r_1$  kao sto i pise u zadatku i to je to. Minimiziramo uz pomoc k tablice i dobijemo not A1 sto je pod E :D.E!

7.	<p>Zadan je bistabil s ulazima <math>A</math> i <math>B</math>, čiji je dijagram prijelaza stanja prikazan na slici desno (navedene pobude su oblika <math>AB</math>). Uporabom bistabila <math>T</math> potrebno je ostvariti ovaj bistabil. Odredite minimalni zapis funkcije ulaza bistabila <math>T</math>.</p> <p>a) <math>\overline{A}B + \overline{Q}A + AB\overline{Q}</math>      c) <math>A\overline{B} + Q</math>      e) <math>A\overline{Q} + BQ</math>  b) <math>A\overline{B} + \overline{Q}</math>      d) <math>Q\overline{B} + \overline{A}\overline{B} + AB\overline{Q}</math>      f) ništa od navedenoga</p>	
----	---	--

7. Imamo bistabil sa ulazima  $A$  i  $B$ . Pisemo tablicu.  $Q$  (trenutno stanje),  $Q_{n+1}$  sljedeće stanje. (ovi kruzici su stanja a ovo na lukovima znaci za koju ce se kombinaciju stanje promjeniti ili ostati isto)

Napisemo tablicu i iz slike gledamo kako se ovo cudo ponasa :D.

Idemo po redovima.

1. red – stanje 0 dolaze 0 0 vidimo da je sljedeće stanje 1, 2. stanje 1 dolaze 00 vidimo da je sljedeće stanje 0, 3. stanje 0 dolazi 01 stanje ostaje 0, 4. red stanje 1 dolazi 01 stanje ostaje 1, 5. red stanje 0 dolazi 10 stanje ostaje 0, 6. red stanje 1 dolazi 10 stanje prelazi u 0, 7. red stanje 0 dolazi 11 stanje ide u 1, 8. red stanje 1 dolazi 11 stanje ostaje u 1. (TABLICA POPUNJENA osim  $T$ )

$T$  bistabil mjenja stanje, ako dodje 1 stanje se mijenja ako dodje 0 stanje ostaje isto i sad popunjavamo (gledamo  $Q$  i  $Q_{n+1}$  (ako se promjeni stanje  $T$  je 1 ako ne  $T$  je 0)), ovo necu pisat jer je jednostavno

$A$	$B$	$Q$	$Q_{n+1}$	$T$
0	0	0	1	1
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	0

Minimiziramo  $T$  i dobijemo rjesenje pod  $D$  sto je točno .D!

8.	Projektirati pretvornik koda koji na 4-bitnom ulazu $a_3a_2a_1a_0$ prima dekadsku znamenku u kodu BCD a na izlazu $r_3r_2r_1r_0$ generira Excess-3 kod znamenke s ulaza. Ako se na ulaz dovede podatak koji ne odgovara BCD kodu znamenke, izlaz nije bitan. Kako glasi minimalni oblik funkcije izlaza $r_2$ ?		
	a) $\bar{a}_2a_0 + \bar{a}_2a_1 + a_2\bar{a}_1\bar{a}_0$	c) $a_2 + a_3\bar{a}_1a_0$	e) $\bar{a}_3 + a_2a_1 + \bar{a}_0$
	b) $\bar{a}_3\bar{a}_2 + \bar{a}_1a_0 + a_1\bar{a}_0$	d) $\bar{a}_3a_2 + \bar{a}_1a_0$	f) ništa od navedenoga

8.Trebamo unjet BCD a da dobijemo Excess -3(sto nam netreba ide u dont care za minimalizaciju).exces 3 dobijemo tako da svakoj brojci dodamo 3(0 kao 3 je kodiran , 1 kao 4 itd.)

Pisemo tablicu:

a3a2a1a0 | r3r2r1r0

0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100
1010	xxxx
1011	xxxx
1100	xxxx
1101	xxxx
1110	xxxx
1111	xxxx

Minimiziramo r2 koristeći jedinice i dont care-ove i dobijemo pod A :D.A!

9.	Uporabom sklopa PLA tipa NI-NI potrebno je ostvariti tri funkcije navedene u nastavku. Koje su minimalno potrebne dimenzije sklopa PLA? $f_1 = \bar{A}\bar{C} + \bar{A}\bar{B}CD + AC\bar{D}$ , $f_2(A,B,C,D) = \sum m(0,1,3,4,11,15)$ , $f_3(A,B,C,D) = \prod M(0,1,3,4,7,12,13,14,15)$ . Naputak: pristupite minimizaciji kao da se radi o minimizaciji višezlazne funkcije.		
	a) $4 \times 6 \times 3$	c) $4 \times 10 \times 3$	e) $4 \times 7 \times 3$
	b) $4 \times 8 \times 3$	d) $4 \times 9 \times 3$	f) ništa od navedenoga

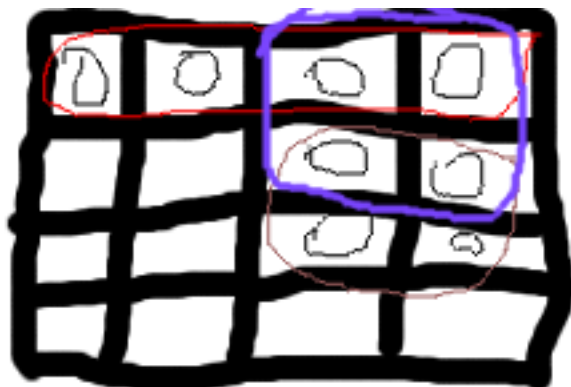
Ove tri funkcije minimiziramo kako je u napatku zadano kao vise izlaznu funkciju .

Minimizaciju viseizlazne funkcije imas u MČ ZBIRKI u 4.23 zadatku

I onda gledamo koliko je potrebno ulaza u svaki NI i ako smo minimizirali vidimo da za 2 funkcije treba po 2 a za jednu treba 3 sto nam daje 7 i rjesenje pod E :D.E!

10.	Funkciju $f(A, B, C, D) = C(\overline{A} + \overline{D}) + \overline{A}(C + D)$ potrebno je minimizirati. Potom je potrebno ukloniti sve hazarde (uz ograničenje da se u istom trenutku može promijeniti samo jedna ulazna varijabla). Koje sume treba još <b>dodati</b> ? <i>Napomena:</i> u kojem obliku treba minimizirati funkciju?					
	a) $B + \overline{C} + \overline{D}$	c) $A + B + D$	e) $\overline{A} + \overline{D}$			
	b) $\overline{A} + C$	d) $\overline{B} + C + \overline{D}$	f) ništa od navedenoga			

10. Napisemo funkciju  $f = \text{not } A \text{ and } C \text{ or } C \text{ and not } D \text{ or not } A \text{ and } D$ . Napisemo tablicu i trazimo maxterme. Maxtermi su 0,4,8,9,11,12,13,15. U k tablici sada unesemo 0 i zaokruzimo gornje 4 i ove srednjo desne 4 . I da bi smo se rijesili hazarda trebamo dodat ove 4 sto su nekom plavom bojom zaokruzene ,pise da se samo 1 ulazna varijabla smije mjenjat pa brijem da to misle da posto nema B u funkciji da se moze mjenjat B i jos jedna iz funkcije, u kojoj se mijenja D , to plavo zaokruzenje nam daje not A or C sto je pod B i to je točno :D.B!



11.	Arhitektura nekog sklopa opisanog jezikom VHDL sastoji se od 3 naredbe prikazane u nastavku. A, B, C i D su ulazi a X, Y i Z izlazi. U nekom trenutku na ulaze se dovede A=0, B=U, C=0, D=1. Što će biti postavljeno na izlaze sklopa?		
	$X \leq (A \text{ AND NOT } B) \text{ OR } (A \text{ AND } B);$		
	$Y \leq (A \text{ OR } B) \text{ AND } (A \text{ OR } C);$		
	$Z \leq (B \text{ OR } C) \text{ AND } (C \text{ OR } D);$		
	a) X=0, Y=0, Z=U	c) X=U, Y=0, Z=U	e) X=U, Y=1, Z=1
	b) X=U, Y=1, Z=1	d) X=U, Y=0, Z=1	f) ništa od navedenoga

11. Ovaj je glupost samo treba znat sa U baratati, pa idemo  
Not U je U , 1 ili U je 1 , 0 ili U je U , 0 i U je 0 , U i 1 je U<- sve sto nam treba.

$X = (0 \text{ and } U) \text{ or } (0 \text{ and } U) = 0$   
 $Y = (0 \text{ or } U) \text{ and } (0 \text{ or } 0) = 0$   
 $Z = (U \text{ or } 0) \text{ and } (0 \text{ or } 1) = U$

Pod A i idemo dalje.A!

12.Ne znam.



13	Pomoću multipleksorskog stabla načinjenog od multipleksora 2/1 bez ulaza za omogućavanje ostvarujemo Booleovu funkciju od četiri varijable, $f(A,B,C,D)$ . Na raspolaganju su nam varijable i komplementi varijabli (to su sve ulazi sklopa), pa funkciju ostvarujemo uporabom trivijalnih rezidualnih funkcija. Sklop koji ostvaruje funkciju $f$ modeliramo strukturno u VHDL-u (multipleksore 2/1 koristimo kao gradivne blokove). Koliko je internih signala potrebno za modeliranje sklopa?				
a) 14	b) 8	c) 10	d) 6	e) 4	f) ništa od navedenoga

13. Koristimo trivijalne rezidualne (rezidualne od 1 varijable – ovdje D) pa nam samo trebaju 3 razine za 3 selekcijska ulaza.

Imamo 2/1 muxeve a trebamo 8 adresnih ulaza pa će prva razina imati 4 muxa druga 2 i treća 1. Između prve i druge razine 4 interna signala jer iz ona prva 4 muxa idu signali u druga dva, i iz druga dva muxa ide po jedan interni signal što nam daje zbroj od 6 i to je pod D.D!

14	Broj AF45 <sub>H</sub> prvo se posmiče udesno logičkim posmakom za 3 bita, a zatim se rezultat te operacije posmiče kružnim posmakom udesno za 5 bitova. Rezultat je:	
a) 815E <sub>H</sub>	c) 47AF <sub>H</sub>	e) 81AE <sub>H</sub>
b) 40AF <sub>H</sub>	d) 8F5E <sub>H</sub>	f) ništa od navedenoga

14. Imamo broj  $AF45(16)=1010111101000101(2)$

prvo pomislimo logički desno za 3 (stavljamo 0 s lijeva i dobijemo)

$0001010111101000$  i sad kružno za 5 bitova, ovih 5 zacrnjenih bitova prebacimo naprijed **010000**  $0010101111$  i dobijemo  $40AF$  što je pod B.B!

15.	Što od sljedećega u jeziku VHDL ne vrijedi?
a) VHDL ne razlikuje velika i mala slova u nazivima signala	
b) interni signali, ako su potrebni, deklariraju se u početnom dijelu arhitekture sklopa	
c) u VHDL-u se ne može modelirati kašnjenje signala	
d) strukturni opis sklopa prepoznajemo po uporabi konstrukta <b>port map</b>	
e) svaki sklop u VHDL-u dovoljno je opisati ili ponašajno ili strukturno	
f) sve od navedenoga vrijedi	

15. rjesenje je pod C, vidi se iz traktora (ja nemam traktor pa nisam vidio.....) C!

16

U nastavku je dan strukturi VHDL-model nekog sklopa (sučelje modela je izostavljeno).

```
architecture struktura of supersklop is
  component sklopNOT is port (x : in std_logic;  y : out std_logic);
  component sklopAND is port (x1,x2 : in std_logic; y : out std_logic);
  component sklopOR  is port (x1,x2 : in std_logic; y : out std_logic);
  signal i : std_logic_vector(0 to 2);
begin
  skop1: entity work.sklopNOT port map (b,i(0));
  skop2: entity work.sklopAND port map (a,i(0),i(1));
  skop3: entity work.sklopAND port map (y => i(2), x1 => c, x2 => i(0));
  skop4: entity work.sklopOR  port map (x2 => i(1), y => d, x1 => i(2));
end struktura;
```

- a)  $d \leq \text{not } b \text{ and } (a \text{ or } c)$
- b)  $d \leq b \text{ and } (\text{not } c)$
- c)  $d \leq a \text{ and } (\text{not } b) \text{ and } c$
- d)  $d \leq (\text{not } a) \text{ or } (\text{not } b)$
- e)  $d \leq a \text{ or } (\text{not } b) \text{ or } c$
- f) ništa od navedenoga

Kako izgleda naredba pridruživanja vrijednosti izlaznome signalu **d** kod ponašajnog modela koji je funkcijski ekvivalentan ovom zadanome?

16. Nisam neki VHDL-aš pa sam dost dugo rješavao zadatak i dobio pod A što je točno . Na Not dolazi B (i njega prenosi interni signal i(0)) i kasnije ćemo imati sa not B. Na prvi I sklop dolazi A i not B(kao i(0)) i izlaz je interni signal i(1), na drugi sklop I dolazi B i not C što je vidljivo iz koda y=izlaz interni signal i(2), x1=ulaz c , x2=ulaz interni signal(i(0)) koji prenosi not B.i na sklop or imamo konačnu funkciju koja glasi  $f = A \text{ and not } B \text{ or not } B \text{ and } C$  , nema ponudjeno nego moramo tražiti. Sva sreća da je prvo ponudjeno točno.A!

17	<p>Multipleksorom 4/1 potrebno je ostvariti funkciju <math>f(A,B,C)=\sum(2,3,5,6)</math>. Označimo s <math>D_0,D_1,D_2,D_3</math> podatkovne ulaze, te s <math>A_1A_0</math> adresne ulaze (indeks 0 označava ulaz najmanje težine). Ako na <math>A_1</math> dovedemo <math>A</math>, a na <math>A_0</math> dovedemo <math>B</math> što treba dovesti na ulaze <math>D_0,D_1,D_2</math> te <math>D_3</math>:</p> <p>a) <math>C,0,0,\overline{C}</math>                      c) <math>\overline{C},C,0,C</math>                      e) <math>C,0,1,C</math>  b) <math>\overline{C},C,0,C</math>                      d) <math>0,1,C,\overline{C}</math>                      f) ništa od navedenoga</p>
----	---

17. pisemo tablicu , i uspoređujemo funkciju sa C, tablicu podjelimo na ova 4 djela i vidimo da kad dobijemo C 01 a funkcija je 00 da C uopce ne utjece na funkciju pa je prvo 0 , samo 1 ponudjeno ima prvo pod 0 točno pa to možemo zaokružiti no idemo dalje da budemo preko 10012301% sigurni :D. C nam je 01 a funkcija 11 , vidimo da c nema utjecaja i tu stavljamo 1,onda imamo c 01 i funkciju 01 i vidimo da su c i funkcija isti pa stavljamo C, i imamo c kao 01 i funkciju kao 10 i vidimo da je to zapravo not C i na kraju smo sva 4 točno izračunali. D!

AB	C	f
00	0	0
00	1	0
-----		
01	0	1
01	1	1
-----		
10	0	0
10	1	1
-----		
11	0	1
11	1	0
-----		

18	<p>Da bismo pomoću multipleksora 4/1 izgradili multipleksor 16/1, koliko nam je potrebno multipleksora 4/1?</p> <p>a) 7                      b) 4                      c) 5                      d) 2                      e) 3                      f) ništa od navedenoga</p>
----	---

18. ovo se mora znati s labosa , 16 adresnih ulaza sto nam daje da prvo trebamo imati 4 muxa pa onda 1 i to je ukupno 5. C!

19.	Na raspolaganju je FPGA sklop kako je prikazano slikom. Što treba upisati u preostali logički sklop kako bi se na izlazu dobila funkcija $f(A,B,C) = \bar{A} + ABC$ ? Prvi (na slici viši) ulaz u CLB je ulaz veće težine. Ponuđena rješenja u CLB se upisuju odozgo prema dolje.
a) 0100	b) 1001    c) 0110    d) 0010    e) 0111    f) ništa od navedenoga

19. Kod ovih gluposti trebamo znati da je gornji ulaz visetezinski.

Gledamo zadnji CLB da vidimo koju funkciju u koncnici daje. Prvo ćemo ulaze označiti sa X i Y. Unosimo u tablicu minterme 0,1,3 i minimiziramo i dobijemo da trebamo dobiti not X i Y. Ako vidimo da je A veće težine od funkcije drugog CLB dolazimo do zaključka da je  $A = X$  i da već imamo not A i da još samo trebamo dobiti da je Y (funkcija drugog CLB) = ABC.

Sad gledamo prvi CLB koji ostvaruje funkciju NI i daje nam

= not B or not C. Na drugi CLB idu A i funkcija prvog CLB (not B or not C)

I sad jednostavnim gledanjem možemo vidjeti kako trebamo iskombinirati A i (not B or not C) da dobijemo ABC. U ABC ide A kao A a funkciju not B or not C trebamo komplementirati. I ta se kombinacija nalazi pod brojem 10 (1 – A je isti, B se komplementira) i dobijemo ABC. I pod D je ponudjeno da je minterm 2 i to zaokružujemo :D.

Ovdje se može i C zaokružiti što kasnijim (valjda, nisam gledao) kombinacijama daje jedno te isto. C ili D !

20.	Ako član $G_i$ u formuli $C_i = G_i + P_i C_{i-1}$ , koja čini temelj realizacije sklopa za izdvojeno generiranje prijenosa ( <i>carry-look-ahead</i> ), poprimi vrijednost 1, što možemo zaključiti? Napomena: $A_i, B_i$ su pri tome ulazi odgovarajućeg potpunog zbrajala.
a) dogodila se pogreška kod generiranja prijenosa b) točno jedan od pribrojnika ( $A_i, B_i$ ) je jednak 1 c) oba pribrojnika ( $A_i, B_i$ ) su jednaka 1 d) $C_{i-1}$ je jednak 1 e) oba pribrojnika ( $A_i, B_i$ ) su jednaka 0 f) ništa od navedenoga	

20. secer na kraju. U zadatku piše da je  $G_i=1$  a ako znamo formulu za  $G_i=A_i$  and  $B_i$  znači da  $A_i$  i  $B_i$  moraju biti 1 da bi uvjet iz zadatka bio ispunjen, vidimo da je pod c ponudjeno da su oba jednaka 1 što i zaokružujemo :D.C!