Tutorial za 2.MI 2006/2007 Grupa A
[by Diablo]

Tutorial i postupci koji bi Vam trebali pomoći kod rješavanja i shvaćanja ovih i sličnih zadataka.
Nisam odgovoran za greške ☺

Skidanjem ovog dokumenta pristali ste donirati autoru bubreg, jetru ili neki drugi organ u slučaju nužde ☺

1. ZADATAK

1. Koji je rezultat simulacije sljedećeg izraza, ako su vrijednosti A='0', B='1', C='U'?

```
f <= A AND NOT B AND NOT C;
```

a) 0
b) 1
c) U
d) nema dovoljno informacija
e) izraz nije moguće izračunati
f) ništa od navedenog

Zadane vrijednosti uvrstite u funkciju pa imamo : f <= 0 * (not 1) * (not U)
Pošto 0 pomnožena s biločim daje 0, rješenje je a) 0

Uzmimo još par primjera, recimo s B-grupe
U B-grupi imamo funkciju f <= 0 + 1 + (not U)
1 plus bilošta je uvijek 1, pa je odgovor u ovom slučaju b) 1

U C-grupi imamo funkciju f <= ((not 0) * 1) + (not U)
Rješenje prve zagrade je 1, a 1 plus bilošta je uvijek 1, pa je rješenje b) 1

U D-grupi imamo funkciju f <= (0 + (not 1)) * (not U)
Rješenje prve zagrade je 0, a nula puta bilošta je uvijek nula, pa je rješenje a) 0

Uglavnom, kad bi imali funkciju gdje bi se 0 zbrajala i množio 1, onda bi funkcija ovisila o U (ili not U), a tablicu za to imate negdje u predavanjima, a ja ju trenutno ne mogu nać :S

2. ZADATAK

2. Kojim se operatorom u VHDL-u obavlja pridjeljivanje vrijednosti nekom signalu?

a) operatorom >=
b) operatorom <=
c) operatorom ==
d) operatorom :=
e) operatorom `put`
f) ništa od navedenog

Ovo je teorijsko pitanje koje bi trebali znati s labosa, rješenje je b) operatorom <=

3. ZADATAK

Za dvije porodice integriranih logičkih sklopova poznati su podaci prikazani u sljedećoj tablici. Ako u nekom složenom sustavu sklopovi porodice P1 pobuđuju sklopove porodice P2, koliko se najviše sklopova porodice P2 može spojiti na izlaz jednog sklopa porodice P1?

	I_{OL} [mA]	I_{IL} [mA]	I_{OH} [μA]	I_{IH} [μA]
P1	16	1,6	400	40
P2	8	0,4	400	20

a) 40
b) 20
c) 10
d) 5
e) 2
f) ništa od navedenog

Nacrtate si dvije kutijice. Neka je na primejr, prva kutijica P1 a druga P2. spajamo izlaze od P1 na P2. znači, IZLAZI (to jest OUT) od P1 se spajaju na ULAZE (to jest IN) od P2. VISOKO (iliti HIGH) spajamo s VISOKIM, NISKO (iliti LOW)spajamo s niskm. I to je cijela filozofija.

Znači, kad spajamo P1 na P2 vrijedi:

$$(P1/P2)_L = (I_{OL})_{P1}/(I_{IL})_{P2}$$

$$(P1/P2)_H = (I_{OH})_{P1}/(I_{IH})_{P2}$$

Zajednički uvjet je manji od ta dva.

Dakle imamo :

P1ol / P2il = 16 / 0,4 = 40

P1oh / P2ih = 400 / 20 = 20

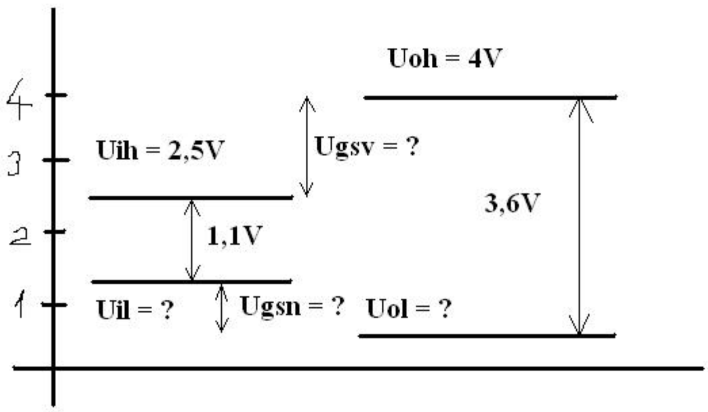
Zajednički uvjet je manji, znači 20 => Rješenje B

Isto vrijedi i kad spajamo P2 na P1, samo okrenete :)

Hvala kolegi Vedaxu

4. ZADATAK

4.	Za neku porodicu logičkih sklopova poznato je sljedeće: U_{OHmin} = 4V, širina zabranjenog područja na izlazu iznosi 3,6V, U_{IHmin} = 2,5V, širina zabranjenog područja na ulazu iznosi 1,1V. Koje su granice istosmjerne smetnje tog sklopa? a) $\|U_{GSV}\|-4V$, $\|U_{GSN}\|-0,4V$, $\|U_{GS}\|-3,6V$ b) $\|U_{GSV}\|-2,5V$, $\|U_{GSN}\|-1,4V$, $\|U_{GS}\|-1,4V$ c) $\|U_{GSV}\|-2,5V$, $\|U_{GSN}\|-1,4V$, $\|U_{GS}\|-2,5V$ d) $\|U_{GSV}\|-1,5V$, $\|U_{GSN}\|-0,4V$, $\|U_{GS}\|-0,4V$ c) $\|U_{GSV}\|-1,5V$, $\|U_{GSN}\|-1V$, $\|U_{GS}\|-1V$ f) ništa od navedenog



Uil = Uih - 1,1V = 1,4V

Uol = Uoh - 3,6V = 0,4V

Ugsv = Uoh - Uih = 1,5V

Ugsn = Uil - Uol = 1V

Ugs se niti netreba računat jer se već sad vidi da je rješenje pod E

|Evo recimo za grupu C još :
|Uih = Uil + 1,1V = 2,5V
|Uoh = Uol + 3,6V = 4V
|Ugsv = Uoh - Uih = 1,5V
|Ugsn = Uil - Uol = 1V
|Ugs se opet netreba ni računati jer se vidi da je rješen pod B
|

5. ZADATAK

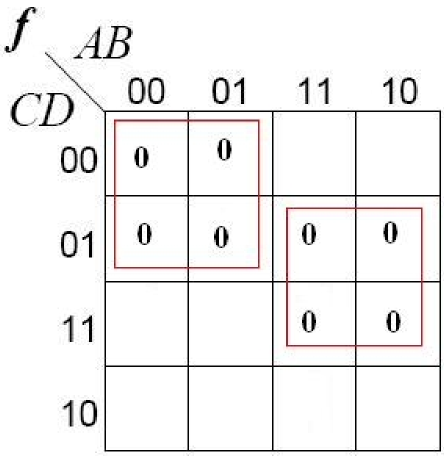
Dunno 

6. ZADATAK

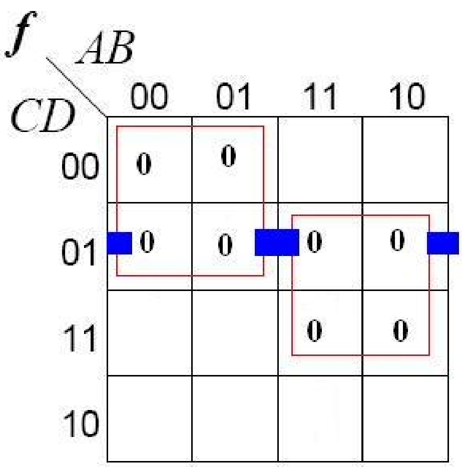
6. Funkcija f(A,B,C,D)=(A+C)(A'+D') direktno je realizirana osnovnim logičkim sklopovima. Na kojem će se prijelazu pobude pojaviti statički hazard?

a) ABCD=0110→ ABCD=1110
b) ABCD=0001→ ABCD=0011
c) ABCD=1101→ ABCD=0101
d) ABCD=0101→ ABCD=1101
e) ABCD=1111→ ABCD=1011
f) ništa od navedenog

Tu si napravite i popunite prvo tablicu ABCD|f i uočite da imate minterme f=m(2,3,6,7,8,10,12,14), odnosno maksterme f=M(0,1,4,5,9,11,13,15)
Sada napravite K-tablicu ovako (napomena – u ovom slučaju gledamo nule jer je zadana produkt suma) :



Hazard se javlja na dodirima zaokruženja. Kod nas je to slučaj na dva mjesta, između polja 5 i 13, te 1 i 9 :



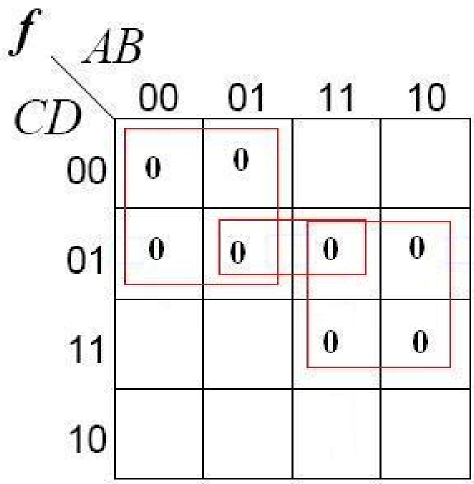
Dakle imamo 4 slučaja, od kojih su 2 točna, a 1 ponuđeni ☺

Slučajevi su
1 -> 9 (0001 -> 1001)
9 -> 1 (1001 -> 0001)
5 -> 13 (0101 -> 1101)
13 -> 5 (1101 -> 0101)

Kod maksterma se hazard javlja iz promjene 0 u 1 (maksterm je 0, pa kad se mjenja 0 u 1)
(usput, kod minterma bi bilo suprotno, on je 1, pa bi se hazard javio kod promjene 1 u 0)

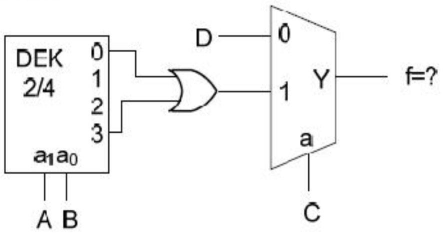
U prvom slučaju vidimo da se prvi bit mijenja iz 0 u 1, to je rješenje točno, ali nije ponuđeno ☺
U drugom slučaju se mijenja 1 u 0, šta je pogrešno
U trećem slučaju se također prvi bit mijenja iz 0 u 1, što je točno i ponuđeno, dakle rješenje je pod D
Četvrti je također pogrešan ko i drugi slučaj.

Mala digresija, da je bilo još pitanje kako se riješiti hazarda, sve što treba napraviti je zaokružiti polja 5 i 13 i time se osigurava da ne nastane hazard



7. ZADATAK

7. Neka funkcija ostvarena je uporabom standardnih kombinacijskih modula. O kojoj se funkciji radi?



a) $\overline{C}D + C(\overline{A}\,\overline{B} + AB)$
b) $ABC + A\overline{B}CD$
c) $AB + AC + BD$
d) $\overline{A}(BD + \overline{C}) + \overline{D}$
e) $ABCD + \overline{A}B\overline{C}D$
f) ništa od navedenoga

Iz dekodera u ILI sklop idu dva izlaza, 0 (00 -> (not A)*(not B)) i 3 (11 -> A*B). Pošto idu na ILI ta dva izlaza se zbrajaju pa imamo (not A)*(not B) + A*B [nazovimo sve to X]. Multipleksor će X propustit ako je C = 1 (C), ILI će propustiti D ako je C = 0 (not C)

Dakle imamo D*(not C) + C*[(not A)*(not B) + A*B] => rješenje A

8. ZADATAK

8. Zadana je funkcija $f(A,B,C,D) = \sum m(2,5,7,8,10,11,13,15)$? Koliko ta funkcija implikanata / bitnih primarnih implikanata?

a) 4 / 0
b) 4 / 4
c) 2 / 2
d) 4 / 1
e) 5 / 3
f) ništa od navedenog

Za ovo je već jedan kolega stavio postupak kako se ovo rješava preko K-tablica, ali moj savjet je da svejedno naučite QM metodu, jer mogu vam stavit minterme m(30,34,41,50,85) a to se baš ne da rješavat preko K-tablica pa ćete svejedno morat znat QM (a i nije tako težak kad se slijedi algoritam).

Dakle, korak prvi :
Sumirajte sve brojeve s istim brojem jedinica u grupe, počevši od najmanjeg broja jedinica do najvećeg :

(2) 0010 - 1 jedinica
(8) 1000
----
(5) 0101 - 2 jedinice
(10) 1010
----
(7) 0111 - 3 jedinice
(11) 1011
(13) 1101
----
(15) 1111 - 4 jedinice

Korak drugi :
Uspoređujte svaki broj iz svake grupe sa svim brojevima iz (isključivo) sljedeće grupe (zato jer se susjedne grupe razlikuju samo u jednoj jedinici) i to tako da kada se dva broja koja uspoređujete razlikuju u samo jednom bitu, umjesto tog bita stavljate _ (underline) a ostale bitove prepišete.
One brojeve koje ste iskoristili (tj našli mu par da se razlikuje u jednoj jedinici) označite ga recimo s x.

(2)...0010 | x | 2,10 | _010 *
(8)...1000 | x | 8,10 | 10_0
----------- |....| 5,7.. | 01_1
(5)...0101 | x | 5,13 | _101
(10).1010 | x | 10,11| 101_
----------- |…| 7,15. | _111
(7) ..0111 | x | 11,15| 1_11
(11).1011 | x | 13,15| 11_1
(13).1101 | x |
----------- |
(15) 1111 | x |

* - uspoređivali smo 2 (iz prve grupe) prvo s 5 (iz sljedeće grupe), no kako se oni ne razlikuju u samo jednom bitu (već u 3), ne možemo ih uspoređivati. Sljedeća usporedba je 2 i 10, i vidimo da se oni razlikuju samo u prvom bitu pa ih zapisujemo u sljedeći stupac na način da na prvi bit stavimo _ (underline) a ostale bitove prepišemo. Pošto smo 2 i 10 uspješno usporedili pokraj njih stavimo iksić (ili kvačicu kako hoćete). Analogno tome se rade i druge usporedbe.
Napomena : 2 se može uspoređivati samo s 5 i 10 (tj brojevi iz prve grupe se mogu uspoređivati samo s brojevima iz druge, jer treća ima tri jedinice što znači da ne postoje brojevi koji bi bili različiti samo u jednom bitu)

Korak treći :
Sada možete uspoređivati samo brojeve kojima se _ (underline) nalazi na istom mjestu, pa tek onda gledati jel se razlikuju u jednom bitu kojeg ćete opet zamjeniti s _

(2) 0010 ..| x | 2,10 | _010 |…| 5,7,13,15| _1_1 **
(8) 1000 ..| x | 8,10 | 10_0 |…| 5,13,7,15| _1_1
----------- |....| 5,7 ..| 01_1 | x |
(5) 0101.. | x | 5,13 | _101 | x |
(10) 1010 | x | 10,11| 101_ |…|
----------- |…| 7,15 .| _111 | x |
(7) 0111.. | x | 11,15| 1_11 |…|
(11) 1011 | x | 13,15| 11_1 | x |
(13) 1101 | x |
----------- |
(15) 1111 | x |

** - jedino smo mogli usporediti 5,7,13,15 i 5,13,7,15, no kako je to isto, možemo prekrižiti jednog (općenito u ovom koraku kad se pojave isti, onda jednog prekrižimo)

Korak četvrti :
Pošto smo usporedili i minimizirali sve što smo mogli, crtamo sljedeću tablicu :

…………..|…2…|…5…|…7…|…8…|…10…|…11…|…13…|…15…

(2,10)……|…x…………………………...x…………………………..
(8,10)……|……………………….x…….x……………………………
(10,11)…..|……………………………x……….x…………………
(11,15)…..|……………………………………...x……………..x…
(5,7,13,15)|…………x…….x…………………………….x………x

Pojašnjenje :
U prvi stupac pišemo sve kombinacije koje nismo iskoristili, tj koje nemaju iksić pokraj sebe (zato stavljamo te iksiće ☺). To su ujedno i primarni implikanti, kojih dakle ima 5.

U prvi redak pišemo brojeve koje smo dobili zadane u funkciji.
Bitne primarne implikante određujemo na način da gledamo koje kombinacije imaju jedine iksić ispod nekog broja u prvom retku.
Dakle (2,10) ima jedina iksić ispod 2, (8,10) ispod 8 i (5,7,13,15) ispod 5,7 i 13.
To su tri komcinacije, odnosno tri bitna primarna implikanta.
Rješenje je pod E) 5/3

Mala digresija :
a) Da je u zadatku još bilo pitanje koliko ima implikanata, to su primarni + iksići (tj svi zapisi koje smo imali)
b) Da je još pisalo na koliko se načina može zapisati rješenje, odgovor je 2, zato jer se na prvi način može zapisati bez kombinacije (10,11) [jer njezine iksiće pokrivaju druge kombinacije], a drugi način je bez (11,15) [iz istog razloga]

P.S.
Možda se čini da je puno texta i da je komplicirano, ali to je samo jer sam pokušao detaljno objasniti, vjerujte mi nije, i ako riješite tri zadatka, shvatit ćete algoritam i nemate frke više.
U četiri koraka dođete do tablice i iz nje samo iščitate sva rješenja za koje vas pita u zadatku uz najmanju mogućnost greške (za razliku od K-tablica). Također, ako vam zadaju neke velike brojeve (ala 40,50,70) ovo je JEDINI postupak rješavanja.

9. ZADATAK

9. Funkcija $f(A,B,C,D)=\sum m(2,4,6,8,9,11)$ realizirana je multipleksorom 2/1, pri čemu je na selekcijski ulaz dovedena varijabla A. Koja se funkcija tada dovodi na prvi podatkovni ulaz multipleksora (ulaz 0)?

a) $\overline{B}\,C + BC$
b) $B\overline{C}D + \overline{B}C$
c) $B + C + D$
d) $(B + C)\cdot\overline{D}$
e) $ABD + \overline{A}BC$
f) ništa od navedenog

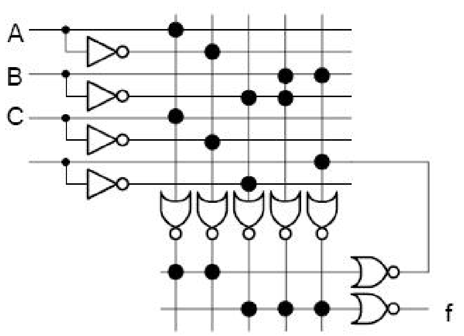
Napravimo ovakvu tablicu :

A.|.B.C.D.|f
0.|..0..0..0.|0
0.|..0..0..1.|0
0.|..0..1..0.|1
0.|..0..1..1.|0
0.|..1..0..0.|1
0.|..1..0..1.|0
0.|..1..1..0.|1
0.|..1..1..1.|0

Objašnjenje :
A je selekcijski ulaz i njega posebno odvojimo. Uzimamo samo prvih 8 kombinacija za koje je A = 0 (jer piše u zadatku da gledamo samo za ove kombinacije)
Sad imamo funkciju f(B,C,D) = m(2,4,6)
Nacrtamo K-tablicu, minimiziramo i dobijemo rješenje C*(not D) + B*(not D).
Izlučimo (not D), pa imamo (B+C)*(not D) => rješenje D

10. ZADATAK

10. Sklopom PLA prikazanim na slici ostvarena je funkcija f. O kojoj se funkciji radi?



a) $f(A,B,C)=\sum m(0,1,3,4,7)$
b) $f(A,B,C)=\sum m(0,3,5,6)$
c) $f(A,B,C)=\sum m(1,2,4,7)$
d) $f(A,B,C)=\sum m(2,5,6)$
e) $f(A,B,C)=\sum m(0,2,4,6,7)$
f) ništa od navedenoga

Gledamo vertikalne linije s lijeva nadesno. Na prvoj vertikalnoj liniji slijeva su označene varijable A i C, koje ulaze u sklop NILI. Znači imamo NOT(A + C). Na drugoj vertikalnoj liniji označeni su not A i not C koji ulaze u NILI, znači imamo NOT((not A) + (not C)).

Na slici se vidi da su ta dva NILI sklopa spojeni na žicu koja ide u treći NILI sklop, dakle imamo NOT(NOT(A+C) + NOT((not A) + (not C))). Sve to zajedno nazovimo recimo X, radi lakšeg snalaženja.
Analogan postupak primjenimo dalje :
Na trećoj vertikalnoj liniji imamo NOT((not B) + (not X)).
Na četvrtoj imamo NOT(B + (not B)) a to je jednako 0 pa ne igra ulogu.
Na petoj imamo NOT(B + X).
Ovi izlazi NILI sklopova ulaze u zadnji NILI koji nam daje funkciju, pa nakon njega imamo NOT(NOT(B+X) + NOT((not B) + (not X))).
Ili grafički :

$$\overline{\overline{A+C}+\overline{\overline{A}+\overline{C}}} = \overline{\overline{A+C}+A*C} = X$$

$$\overline{\overline{\overline{B}+\overline{X}}+\overline{B+X}} = \overline{B*X+\overline{B+X}} = f$$

Eh, dalje sam ja rješavao tako da sam uvrštavao kombinacije u prvi red da dobijem X, i onda u drugi da dobijem f, pa vidim kad će mi ispast f=1.

Recimo uvrstimo 0 (ABC=000)
X će ispast 0.
f će ispast 0.
Dakle 0 nije minterm, pa odma otpadaju rješenja a, b, e

Uvrstimo sad 1 (ABC=001)
X će ispast 1.
f će ispast 1.
Dakle 1 je minterm, što znači da je konačno rješenje C, pošto je A otpalo maloprije :)

11. ZADATAK

11. Što nam je još potrebno za izgradnju potpunog zbrajala ako na raspolaganju imamo dva poluzbrajala?

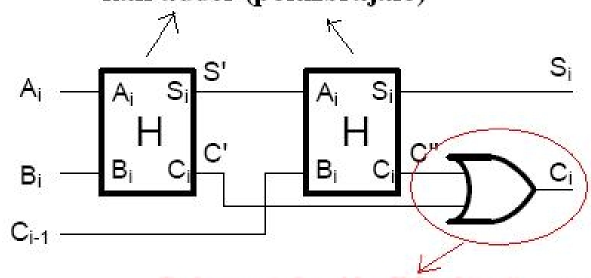
a) tri invertora
b) jedan logički sklop ILI
c) jedan logički sklop I
d) dva invertora
e) jedan logički sklop NILI
f) ništa od navedenog

Binarno zbrajalo

- sklopovska izvedba zbrajanja tri bita
  ~ *potpuno zbrajalo* (engl. full-adder):
  kaskadiranje dva poluzbrajala!

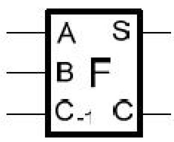
$$S' = A_i \oplus B_i$$
$$C' = A_i \cdot B_i$$
$$S_i = S' \oplus C_{i-1}$$
$$C_i = S' \cdot C_{i-1} + C'$$
$$= C'' + C'$$

half adder (poluzbrajalo)



Treba nam jedan sklop ILI za bit prijenosa

Rješenje B



-> **full adder (zbrajalo)**

12. ZADATAK

12. Neki digitalni sklop radi s naponima -2V i -4V. Neka su ulazi sklopa *A* i *B*. Odziv sklopa za sve kombinacije napona prikazan je tablicom. Koju funkciju *f* ostvaruje taj sklop u negativnoj logici?

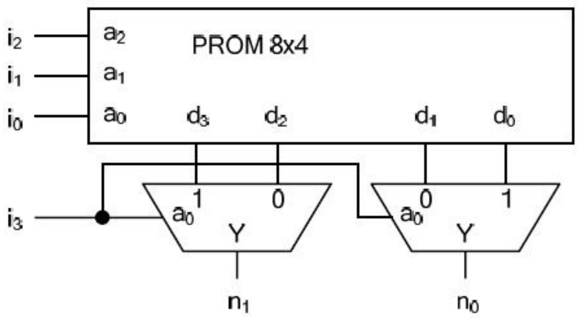
A	*B*	*f*
-2V	-2V	-4V
-2V	-4V	-4V
-4V	-2V	-4V
-4V	-4V	-2V

a) NI
b) NILI
c) I
d) ILI
e) Ex-ILI
f) ništa od navedenoga

Ovdje -4V zapravo predstavlja 1, a -2V predstavlja 0.
Kad ispišete tablicu s novim vrijednostima(1 i 0), očito je da je riječ o funkciji NI, dakle rješenje A

13. ZADATAK

Sklopom temeljenim na ispisnoj memoriji potrebno je realizirati funkciju *P*(*i*) koja za zadani *i* vraća *i*-ti element iz niza {0,1,2,3,3,2,1,0,0,1,1,3,3,2,2,1} (numeracija kreće od nule). Što treba upisati u ispisnu memoriju? U ponuđenim odgovorima prikazan je sadržaj po memorijskim lokacijama, počev od adrese 0, u heksadekadskom obliku, pri čemu je bit d_3 bit najveće težine.



a) 1, B, E, 8, 1, 7, E, 9
b) 0, C, 6, F, F, 3, 9, 4
c) 0, C, A, F, F, 3, 5, 8
d) 0, 3, 9, F, F, C, 6, 1
e) 0, 3, 5, F, F, C, A, 1
f) ništa od navedenoga

Tu se uvijek crta ovakva tablica :

i3 i2 i1 i0 | d3 d2 d1 d0
----
0 0 0 1 |
0 0 1 0 |
0 0 1 1 |
0 1 0 0 |
0 1 0 1 |
0 1 1 0 |
0 1 1 1 |

E sad, i3 nam predstavlja selektivni ulaz za multipleksore.
Ako je i3 jednako 0 (prvih 8 kombinacija u tablici), onda propušta na prvom multipleksoru d2, a na drugom d1.
Kad će i3 biti jednako 1 (drugih 8 kombinacija u tablici), onda će propuštat d3 i d0.

U prvom slučaju, d2 i d1 je predstavljati izlazne bitove n1 i n0, a oni su binarna kombinacija pojedinog zadanog elementa iz niza zadanog u zadatku (tj, kad su n1n0 = 00, to će biti 0 iz niza, n1n0 = 11, to će biti 3 iz niza itd).

Pa počnimo!

i3 i2 i1 i0 | d3 d2 d1 d0
----
0 0 0 1 ....|........ 0 0......... => i3 je 0,na d2d1 pišemo binarnu kombinaciju prvog elementa (0)
0 0 1 0 ....|........ 0 1......... => i3 je 0,na d2d1 pišemo binarnu kombinaciju drugog elementa (1)
0 0 1 1 ....|........ 1 0......... => i3 je 0, na d2d1 pišemo binarnu kombinaciju trećeg elementa (2)
0 1 0 0 ....|........ itd.......... => shvatili ste poantu pa da ne pišem sad sve :P
0 1 0 1 ....|
0 1 1 0 ....|
0 1 1 1 ....|

Kad ste popunili tablicu za i3 = 0, onda popunjavate na isti način za i3 = 1, samo upisujete sljedeće elemente na mjesta d3 i d0 i to od vrha J

i3 i2 i1 i0 | d3 d2 d1 d0
----
0 0 0 0 ....| ...0 0 0 0...... => i3 = 1, na d3d0 pišemo binarnu kombinaciju devetog elementa (0)
0 0 0 1 ....| ...0 0 1 1 ......=> i3 = 1, na d3d0 pišemo binarnu kombinaciju desetog elementa (1)
0 0 1 0 ....| ...0 1 0 1 ......=> i3 = 1,na d3d0 pišemo binarnu kombinaciju 11. elementa (1)
0 0 1 1 ....| ......1 1 ....=> shvatili ste poantu pa da ne pišem sad sve :P
0 1 0 0 ....| 1 1
0 1 0 1 ....| 1 0
0 1 1 0 ....| 0 1
0 1 1 1 ....| 0 0

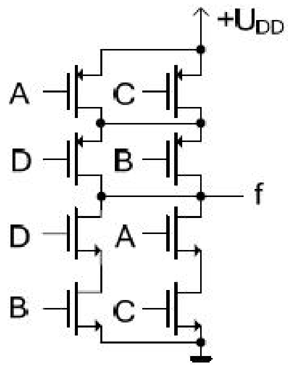
I sad na kraju samo iščitate hexadekatske vrijednosti iz d3d2d1d0 :

i3 i2 i1 i0 | d3 d2 d1 d0
----
0 0 0 0 ....|... 0 0 0 0 .....=> 0
0 0 0 1 ....|... 0 0 1 1 .....=> 3
0 0 1 0 ....|... 0 1 0 1 .....=> 5
0 0 1 1 ....|... 1 1 1 1 .....=> F
0 1 0 0 ....|... 1 1 1 1 .....=> F
0 1 0 1 ....|... 1 1 0 0 .....=> C
0 1 1 0 ....|... 1 0 1 0 .....=> A
0 1 1 1 ....|... 0 0 0 1 .....=> 1

Rješenje je pod E) 035FFCA1

14. ZADATAK

14. Funkcija *f* izvedena je u CMOS tehnologiji. O kojoj se funkciji radi?



a) $\overline{A}\,\overline{D} + \overline{B}\,\overline{C}$

b) $\overline{A}\,\overline{B} + \overline{C}\,\overline{D}$

c) $(\overline{A} + \overline{C})(\overline{B} + \overline{D})$

d) $AB + CD$

e) $(\overline{A} + \overline{D})(\overline{B} + \overline{C})$

f) ništa od navedenoga

Ovo je bilo u prvom međuispitu, ali da ponovimo :
- Gornja polovica sheme kad je iščitamo nam daje rješenje f, a donja polovica (not f).
- Ako na izlazu imamo invertor, onda je obrnuto, što znači da u prvom slučaju samo iščitavate gornju polovicu, a ako imate invertor onda samo donju.
- U gornjoj polovici iščitavate komplementirane varijable (znači (not A), (not B) etc), a u donjoj obične (nekomplementirane).
- Ako su tranzistori u seriji, to predstavlja množenje, a ako su u paraleli onda zbrajanje

Pa slijedimo algoritam :
- Na izlazu nema invertora, znači iščitavamo gornju polovicu sheme
- To ujedino znači da iščitavamo komplementirane verijable
- Pošto su A i C u paraleli (zbrajanje), imamo (not A) + (not C)
- Oni su ujedino u seriji sa paralelom od B i D, znači [(not A) + (not C)]*[(not B) + (not D)]
- Dakle rješenje je pod C

15. ZADATAK

Dunno, mrzim prokleti VHDL