

Digitalna logika – Završni ispit

02. 02. 2006.

02.02.2006.

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

B-1

Završni ispit iz Digitalne logike

Grupa B

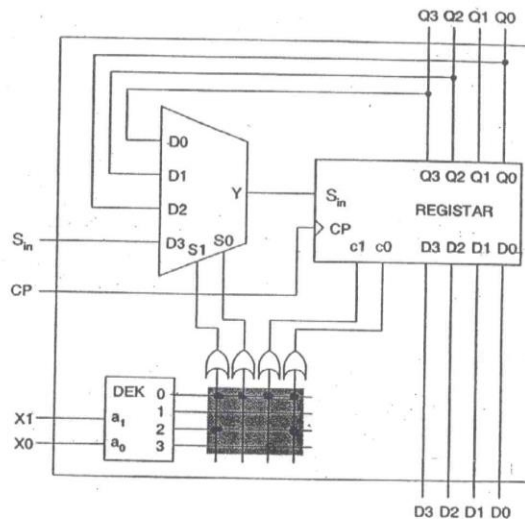
1. Na raspolaganju je posmačni registar koji ovisno o ulazima C_1 i C_0 obavlja jednu od operacija, prema tablici 1-1. Uporabom ovog registra i multipleksora želi se ostvariti registar čija je funkcija definirana ulazima X_1 i X_0 , prema tablici 1-2. Kako treba programirati PROM prikazan na slici da bi se dobio traženi registar?

C_1	C_0	Opis
0	0	Ništa (NOP)
0	1	Posmak u desno, punjenje sa S_{in}
1	0	Paralelni upis
1	1	Posmak u lijevo, punjenje sa S_{in}

Tablica 1-1

X_1	X_0	Opis
0	0	Posmak u lijevo, punjenje sa S_{in}
0	1	Ništa
1	0	Rotacija u desno
1	1	Paralelni upis

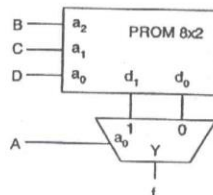
Tablica 1-2



Kao rješenja je ponuđen sadržaj memorije, počev od najniže lokacije. Lijevo na slici nalazi se bit najveće težine. Ukoliko vrijednost nekog bita u memoriji nije bitna za rad sklopa, taj bit postavite na 0.

- a) 5,2,7,13 b) 1,3,0,14
c) 2,2,2,2 d) 15,0,9,2
e) 13,7,13,5

2. Funkciju $f(A, B, C, D) = \sum m(0, 2, 4, 5, 6, 9, 11, 12, 15)$ potrebno je ostvariti sklopom prikazanim na slici. Koje podatke treba upisati u memoriju, počev od najniže lokacije?



- a) 1,1,2,0,3,1,2,3 b) 1,2,1,2,3,1,1,2
c) Funkciju nije moguće ostvariti d) 0,3,1,1,2,2,1,3
e) 3,1,3,2,1,2,3,0

1. Analizirajte zadatak po redu za svaki ulaz ovog sklopa:

Za ulaz 00 se na izlazu dekodera aktivira **samo** vod 0 (prva žica), I po tome vodu označujete koji bitovi za ovu funkciju trebaju biti aktivni. U ovom slučaju treba ostvariti funkciju posmaka u lijevo i punjenja sa ulazom sklopa (S_{in}). Krenimo prvo od multipleksora. Multipleksor ovisno o adresnim ulazima (ova dva dolje) bira koji će podatkovni ulaz (ovi lijevo) proslijediti na izlaz Y, i to tako da binarno očita adresne ulaze i prosljeđuje podatkovni ulaz s tim indeksom. U ovom slučaju treba proslijediti S_{in} , pa na podatkovne ulaze trebamo dovesti 3, tj 11_2 znači prva 2 „križanja“ na vodu 0 označimo. Registar pak treba ostvariti posmak ulijevo, što će reći da se svaki bit miče za 1 u lijevo, q_3 se briše, a na mjesto q_0 dolazi ulaz S_{in} , nađemo ekvivalentnu funkciju u tablici funkcija registra: njezina pobuda je također 3, znači sva „križanja“ na vodu su označena kak je i kolega koji je ovo uploadao to i označio.

01-ništa – sad je svejedno kaj dovedemo iz multipleksora, zato što posmačni registar to neće koristiti, ali u zadatku piše da ako neki bit nije važan, ostavimo ga u 0, što znači da su prva dva križanja voda 1 prazna, kao i druga 2, jer je funkcija „ništa“ registra također 00.

Ekvivalentno ovome možete sada riješiti ostatak zadatka, s tim da pazite da rotacija znači da se ovaj zadnji bit kaj bi se inače brisao dovodi na S_{in} i tako upisuje na prvo mjesto, znači njega trebate odabrati muxom. Primijetite još da je na vodu 3 označeno i 3. „križanje“!

Nakon što ste riješili zadatak, očitajte stanje ovog ili-polja, i to tako da binarno očitajte svaki vodoravni vod. Označena „križanja“ su u 1, neoznačena u 0. znači vodovi po redu su:

0: 1111 = 15

1: 0000 = 0

2: 1001 = 9

3: 0010 = 2

I eto vam odgovora na pitanje. Primijetite još da je dovoljno da riješite samo vod 0 i očitajte ga, jer svi odgovori počinju različitim brojem, znači čim označite cijeli prvi vod, očitajte da je to 15, samo odgovor d počinje s 15, pa on mora bit točan.

2. Prom je ista stvar ko i ovaj dekodere i programirljivo ili-polje koje sam opisao maloprije, samo što je ovaj drugih dimenzija. Ima 3 ulaza, što znači 8 vodova, a samo dva izlaza. Napišite tablicu ove funkcije:

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

1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

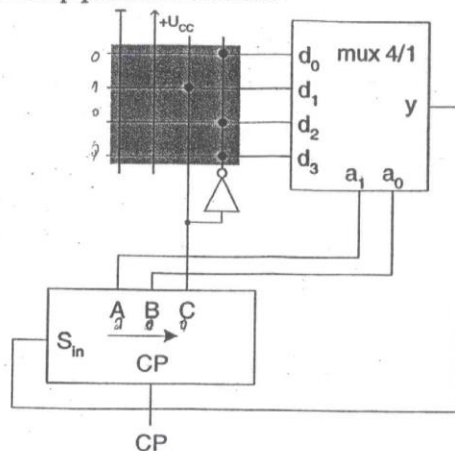
A je adresni ulaz multipleksora, znači kad je A u 1, dovest će se izlaz PROM-a d_1 , a kad je u 0, dovest će se izlaz d_0 . Zbog ovoga je najbolje gledati gornju tablicu u 2 dijela: kad je A u 0 i drugi dio kad nije. A=0 će odgovarati izlazu d_0 , prema tome, u desnom okomitomvodu PROM-a treba označiti ona „križanja“ u kojima je funkcija aktivna kada je A=0, znači, križanja s vodovima 0,2,4,5,6, a za A=1 treba označiti križanja 9,11,12 i 15. Znači sadržaj proma je sljedeći:

	d_1	d_0
0		1
1	1	
2		1
3	1	
4	1	1
5		1
6		1
7	1	

Iliti: 1,2,1,2,3,1,1,2 – odgovor b.

3. U kojem ciklusu broji sklop prikazan na slici?

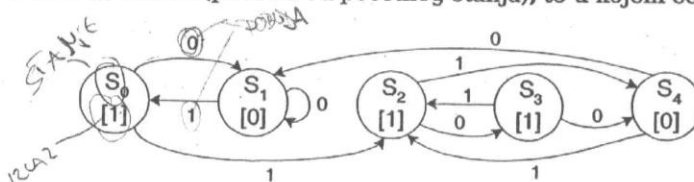
A
1



- a) 0,4,6,7,3,5,2,1
b) 4,2,1,5
c) 1,0,2,3,4,5,6,7
d) 3,7,2
e) 0,4,2,5,6,3,1

4. Stroj s konačnim brojem stanja realiziran je prema dijagramu prikazanom na slici. Ako se početno stroj nalazi u stanju S0, te se na ulaz dovede slijed 0,1,1,1,0, što će stroj generirati na izlazu (počevši od početnog stanja), te u kojem će stanju ostati?

1 S 1
0 S 0
1 S 2



- a) izlazi: 1,0,0,1,0,1, stanje S2
b) izlazi: 1,0,0,1,1,0, stanje S4
c) izlazi: 1,0,0,1,1,1, stanje S0
d) izlazi: 1,0,0,0,0,1, stanje S3
e) Izlazi: 1,0,1,1,0,0, stanje S1

5. Na raspolaganju je T bistabil. Njegovom uporabom, te uz minimalni utrošak osnovnih logičkih sklopova, projektirajte xy bistabil, čije je ponašanje definirano tablicom.

x	y	Q_{n+1}
0	0	0
0	1	Q_n
1	0	$\overline{Q_n}$
1	1	1

- a) $T = X \cdot \overline{Q} + \overline{Y} \cdot Q$
b) $T = \overline{Q} \cdot X + \overline{Q} \cdot Y + \overline{Q} \cdot \overline{X} \cdot \overline{Y}$
c) $T = \overline{Q} \cdot X + \overline{X} \cdot Y + \overline{Q} \cdot X \cdot \overline{Y}$
d) $T = \overline{Q} \cdot X + \overline{Q} \cdot Y$

b) $T = \overline{Q} \cdot X + \overline{Q} \cdot Y + \overline{Q} \cdot \overline{X} \cdot \overline{Y}$
d) $T = \overline{Q} \cdot X \cdot Y + Q \cdot X + X \cdot \overline{Y}$

3. Možete pretpostaviti bilo koje početno stanje, ja obično krećem od 000: A i B se dovode na ulaze multipleksora, znači prosljeđuje se izlaz d_0 , na kojemu je C', iliti 1. Posmačni registar sad miče A i B za 1 bit u desno, i puni prvi bit s izlazom muxa, u ovom slučaju 1, znači sljedeći broj je $100_2 = 4_{10}$. Sada su A i B 10, što znači da se u muxu prosljeđuje izlaz d_2 , tj. opet C' – opet 1. Pomičemo A i B udesno, punimo s 1, sljedeći broj je dakle $110_2 = 6_{10}$. Ovo nam je već dovoljno da zaključimo da je točan odgovor a.
*Napomena: prva dva okomita voda u PROM-u su uzemljenje i napajanje, iliti logička 0 i logička 1.

4. Brojevi u uglatim zagrada su izlazi, a oni na strelicama su pobude. Krećemo iz stanja S0, koje ima izlaz [1]. Po zadanom slijedu koristimo pobude, i tako prelazimo u sljedeća stanja. Prva pobuda je 0, što nas prenosi u stanje S1 s izlazom [0]. Sljedeća pobuda je 1, vraća sklop u stanje S0 s izlazom [1]. Zatim pobuda 1 stavlja sklop u stanje S2, izlaz [1], nova 1 prebacuje u stanje S4, izlaz [0], te zadnja pobuda 0 prebacuje sklop u stanje S1, koje ima izlaz [0], tj. kraće:

$$S0[1] + 0 > S1[0] + 1 > S0[1] + 1 > S2[1] + 1 > S4[0] + 0 > S1[0]$$

I odgovor je 1,0,1,1,0,0 (e)

5. Kao prvo, T- bistabil ima samo jedan ulaz i CP, na svaku 1 na ulazu on mijenja stanje, a na 0 ostaje isti.

Tablica XY bistabila (skupa s prethodnim stanjem) je sljedeća:

Q_n	X	Y	Q_{n+1}	T
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

Stupac T se dobije tako da uspoređujete Q_n sa Q_{n+1} – ako ima promjene, T je u 1, ako nema, T je u 0.

Sad napišete T kao funkciju Q_n , X i Y (suma minterma):

$$T = Q_n'XY' + Q_n'XY + Q_nX'Y' + Q_nXY', \text{ pa minimizirate i dobijete:}$$

$$T = Q_n'X + Q_nY'$$

(a)

- ✗ Uporabom minimalnog broja T bistabila s asinkronim ulazom za brisanje želi se projektirati asinkrono binarno brojilo koje broji u ciklusu duljine 12. Koje je stanje u tu svrhu potrebno dekodirati? Svi ulazi za brisanje spojeni su zajedno.

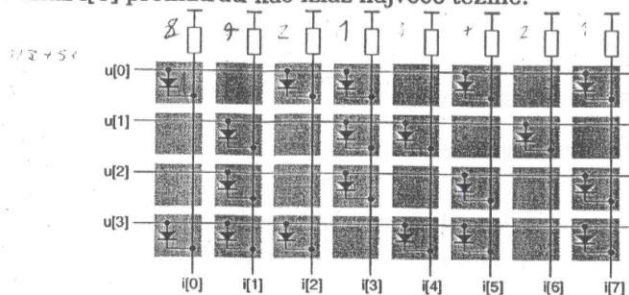
a) 0
b) 12
c) 14
d) 15
e) 13

BRISANJE X STANJE
POSTAVLJANJE (X-1) STANJE

7. Permanentna memorija ostvarena je diodnim poljem, prema slici. Koji je sadržaj memorije? Izlaz i[0] promatrati kao izlaz najveće težine.

B✓

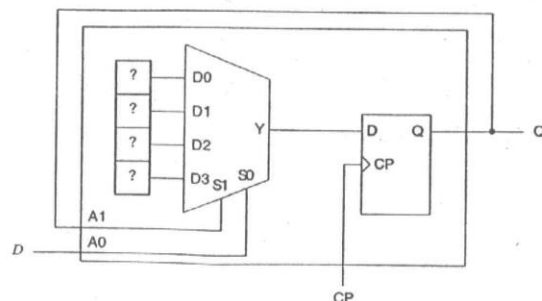
1



a) 4A,A5,AA,12
b) B5,5A,55,ED
c) 23,17,2E,35
d) A0,ED,54,31
e) 14,73,77,51

8. Programirajte logički blok FPGA sklopa temeljen na preglednoj tablici, tako da ostvarite D bistabil. Što se upisuje u preglednu tablicu (odozgo prema dolje)?

1



→ a) 0,1,0,1
b) 1,1,0,0
c) 0,0,1,1
d) 1,0,1,0
e) 0,1,1,0

9. Prednost $2^{1/2}D$ organizacije memorije u odnosu na 2D organizaciju je:
- a) Povećanje disipacije
b) Produljenje vodova riječi
c) Produljenje vodova bita
d) Skraćivanje vodova bita
e) Skraćivanje vodova riječi

6. Ima 12 stanja, znači zadnje stanje je 11 (počinjemo od 0). Imamo ulaz za **brisanje**, znači kad se on aktivira, svi bistabili će past u 0. Sad bi mi svi sretno rekli, pa dobro, kad dođe u zadnje stanje, kodiram ga da ode u 0000. E pa nije tak. Problem je što imamo asinkroni ulaz za brisanje, znači on neće čekat do sljedećeg CP-a da pređe u sljedeće stanje, neg će sve zbrisat odmah, tak da će brojilo u biti na tren bit u stanju 11, i odma preć u 0, a na sljedeći CP će već preć u 1. Upravo zato što ulaz za brisanje radi asinkrono, tj. ne na pobudu CP, kak treba radit brojilo, mi ćemo dopustit da brojilo ode na tren u stanje 12, pa kad ga sklop za brisanje detektira, on odma brojilo prebaci u stanje 0. Znači trebamo dekodirati stanje 12. Odgovor je b)

Ako vam se ne daj Bože dogodi da u zadatku kaže da ima sklop za **postavljanje**, to trebate shvatit kao da se podrazumijeva da je zadnje stanje 1111, znači kad bi imali brojilo sa 12 stanja, zadnje stanje (11) na bi bilo 1011, nego 1111, znači mi bi onda morali dekodirat stanje 11, da sklop za postavljanje kad detektira 1011, odma prebaci to u 1111.

Može se dogodit i da trbate sami projektirat taj sklop za brisanje ili postavljanje. Ja ću ovdje objasniti sam kaxe radi brisanje za sklop sa 12 stanja: znači rekli smo kad detektira stanje 12, iliti 1100, mora preć u 0. Znači izlaze Q3, Q2, Q1', i Q0' spojite na I sklop, i dobit ćete sklop za brisanje. Važno je još primijetit da su Q3 i Q2 prvi puta u 1, kada se dođe u stanje 12, tako da je dovoljno da samo Q3 i Q2 dovedete na I sklop i spojite na ulaze za brisanje. Još samo treba paziti da li ulazi za brisanje djeluju kada se na njih dovede 0 ili jedinica, ako djeluje 0, samo komplementirate I sklop, tojest spojite Q3 i Q2 na NI sklop. Znači ak je zadatak da napišete funkciju za sklop za postavljanje, vjerovatno će vam bit odgovori ovog kraćeg oblika. I odgovor je onda Q3Q2.

7. Ovaj zadatak je fakat lagan, polja koja imaju diodu su u 1, a ona koja nemaju su u 0, i onda binarno očitajte sadržaj memorije odozgora prema dolje, pretvorite u hexadecimalno i voila! U slučaju da dobijete memoriju s mosfetima, slučaj je obrnut, polja u kojima su mosfeti su u 0, a prazna su u 1.

8. Napišeš tablicu D-bistabila, on je u 1 kad mu je ulaz u 1, a u 0 kad mu je ulaz u 0.

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

Q_n i D se dovode na adresne ulaze muxa, a podatkovni su upravo ekvivalentni s Q_{n+1} prema tome, samo prepisete Q_{n+1} u LUT.

9. Ovaj neam pojma, pogledajte si u zbirci il negdje ☺, al možete si odma eliminirat 3/5 odgovora, jer oni nisu prednosti, tj. odgovor može bit samo d ili e.

10. Ovaj imate u knjizi digitalni sustavi, negdje pri početku DA pretvorbe. To je naš objašnjeno s Theveninovim teoremom, a čim sam to vidio, pao mi je mrak na oči, al zapamtil sam formulu jer je prilično lagana. Uglavnom, bit je u tome da bit najveće težine (kada je aktivan) ima najveći napon, onaj za jednu težinu manji ima duplo manji napon, treći duplo manje od drugog, pa tak sve do zadnjeg. Bit najveće težine ima napon $U_{ref}/2$.

Sada imamo podatak 01110, znači to je $U_{ref}/4 + U_{ref}/8 + U_{ref}/16 = 7 \Rightarrow U_{ref}=16V$. I sad izračunate po tome napon za podatak 11011: $U_{ref}/2 + U_{ref}/4 + U_{ref}/16 + U_{ref}/32 = 13.5V$
(a)

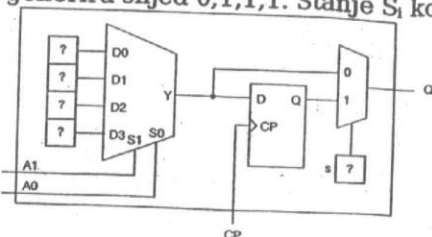
11. Ni ovaj ne znam, pogledajte si ga negdje, al točan odgovor je e

12. Napišimo tablicu stanja, izlaz Q1 je za T-bistabil, a Q2 za D, primijetite da se na ulaz D-a dovodi Q' od T, a ne Q.

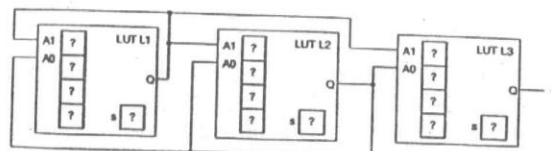
n. stanje		n+1. stanje	
Q ₁	Q ₀	Q ₁	Q ₀
0	0	0	1
0	1	1	1
1	0	1	0
1	1	0	0

Siguran start sklopa znači da sklop prolazi kroz određeni ciklus stanja, te ako se dogodi da sklop upadne u neko stanje koje nije u zadanom ciklusu, njegova funkcija ga vraća nazad u ciklus. U ovom sklopu nije zadan glavni ciklus, no možemo očitati da postoje dva nepovezana ciklusa 00->01->11->00 i 10->10. Bez obzira koji je od ova dva ciklusa glavni, sklop sigurno nema siguran start. Ako je glavni ciklus 0->1->3->0, u slučaju da sklop upadne u stanje 2, ne može se vratiti iz njega, budući da stanje 2 uvijek vraća sklop u U ovom sklopu nije zadan glavni ciklus, no možemo očitati da postoje dva nepovezana ciklusa 00->01->11->00 i 10->10. Bez obzira koji je od ova dva ciklusa glavni, sklop sigurno nema siguran start. Ako je glavni ciklus 0->1->3->0, u slučaju da sklop upadne u stanje 2, ne može se vratiti iz njega, budući da stanje 2 uvijek vraća sklop u stanje 2. A ako je 2 glavni ciklus, ako sklop upadne u neko drugo stanje slučajno, nikada se neće vratiti u stanje 2, jer su ostala stanja također zatvoren ciklus.

14. Na raspolaganju su tri logička bloka FPGA sklopa prikazana na slici 14-1, spojenih prema slici 14-2. Konfigurirajte blokove tako da se dobije sklop koji na izlazu generira slijed 0,1,1,1. Stanje S_1 kodirajte binarnom reprezentacijom od 'i'.



Slika 14-1



Slika 14-2

- | | |
|---|---|
| a) $S_1=1$, LUT1=0,1,1,0
$S_2=1$, LUT2=1,0,1,0
$S_3=0$, LUT3=0,1,1,0 | b) $S_1=0$, LUT1=0,0,0,0
$S_2=1$, LUT2=0,0,0,0
$S_3=1$, LUT3=0,0,0,0 |
| c) $S_1=1$, LUT1=1,1,0,0
$S_2=0$, LUT2=0,0,1,1
$S_3=1$, LUT3=0,1,1,1 | d) $S_1=1$, LUT1=0,1,1,0
$S_2=1$, LUT2=1,0,1,0
$S_3=0$, LUT3=0,1,1,1 |
| e) $S_1=1$, LUT1=0,1,0,0
$S_2=1$, LUT2=0,0,1,0
$S_3=0$, LUT3=0,1,1,1 | |

15. Blok PROCESS nekog bistabila modeliranog VHDL-om prikazan je u nastavku. Koji je od ponuđenih odgovora točan za taj bistabil?

```
PROCESS (cp, a, b, x, qint)
BEGIN
```

```
  if (b='0') then qint <= '0'; elsif (falling_edge(cp)) then
    if (a='0') then qint <= '1'; else qint <= qint XOR x; end if;
  end if;
```

```
END PROCESS;
```

- | | |
|------------------------------------|---------------------------------------|
| a) a i b djeluju sinkrono | b) b djeluje sinkrono, a asinkrono |
| c) a i b djeluju asinkrono | d) sklop reagira na rastući brid cp-a |
| e) a djeluje sinkrono, b asinkrono | |

16. Blok PROCESS nekog bistabila modeliranog VHDL-om prikazan je u nastavku. O kojem se bistabilu radi? Signal qint direktno se preslikava na izlaz sklopa.

```
PROCESS (cp)
```

```
BEGIN
```

```
  if (falling_edge(cp)) then
    qint <= x;
  end if;
```

```
END PROCESS;
```

- | | |
|---------------------------------------|--|
| a) T bistabil okidan padajućim bridom | b) D bistabil okidan rastućim bridom |
| c) T bistabil okidan rastućim bridom | d) JK bistabil okidan padajućim bridom |
| e) D bistabil okidan padajućim bridom | |

14. Imamo slijed od četiri stanja, znači na neku foru treba napraviti da taj sklop skuži koliko je stanja prošlo (čitaj: napraviti brojilo). A budući da je ciklus od 4 stanja, treba nam brojilo od 4 stanja, znači trebaju nam nekakva dva bistabila. L3 daje izlaz funkcije, znači L1 i L2 trebaju programirati tako da njih dvojica ostvare brojilo. Da vidimo mi sad tablicu stanja brojila:

Q_n		Q_{n+1}	
Q_1	Q_0	Q_1	Q_0
0	0	0	1
0	1	1	0
0	0	1	1
0	1	0	0

S tim da je Q_1 izlaz LUT1, a Q_0 izlaz LUT2. Ako pratite pažljivo što treba biti izlaz multipleksora u svakom LUT-u, vidjet ćete da se u svaki LUT upisuju upravo odgovarajuće vrijednosti $(n+1)$. stanja.

Kad su izlazi Q_1 i Q_0 oba u 0, mux prosljeđuje adresu D0. Budući da sljedeće stanje treba biti 01, prva adresa LUT1 će biti 0, a LUT2 1. 00 ćemo tretirati kao prvo stanje, znači kada se 00 dovede u LUT3, izlaz treba biti 0. Dalje možete i sami zaključiti da su Q_1Q_0 ono što se upisuje u LUT1 i LUT2, a izlazi 0111 se upisuju u LUT3. S je u jedinici u LUTovima 1 i 2, jer je potreban bistabil da pamti prethodno stanje, kako bi se realiziralo brojilo, dok LUT3 samo djeluje kao multipleksor, tako da mu ne treba bistabil, $S3=0$.

Odgovor: d)

15. Što znači da je varijabla sinkrona? To znači da njena funkcija ovisi o pobudi na CP-u, CP "sinkronizira sklop" tako da odredimo da svaki element tog sklopa funkcionira samo kad dobije određenu pobudu na ulazu CP, (ta pobuda može biti 1, 0, rastući ili padajući brid), te svaki element počinje funkcionirati u isto vrijeme, zato kažemo da je sinkron. Ako ne ovisi o cp-u onda očito nije sinkron, tj. asinkron je. U ovom zadatku vidimo da se s obzirom na b qint mijenja prije provjere cp-a (prije neg kaj piše `elsif falling_edge(cp)`), znači b je neovisan o cp, b je asinkron, a a je očito sinkron, jer je unutar uvjeta (`elsif falling_edge(cp)`).

16. Za početak falling znači padajući ☺ A qint je izlaz, koji ovisi samo u pobudi x, T-bistabil ovisi i o prethodnom stanju i o pobudi, tak da on nije, a D-bistabil ovisi samo o pobudi (ak je D 0, izlaz je 0, ak je D 1, izlaz je 1). Tako da se očito radi o D-bistabilu okidanom padajućim bridom. Odgovor: e)

17. Arhitektura nekog sklopa koji ima ulaze x, y, z te izlaz o (svi tipa `std_logic`) modelirana je VHDL-om prikazanim u nastavku. O kojem se sklopu radi?

A $o \leq x$ WHEN $z = '0'$ ELSE y ;

1 ☒ a) Multipleksor 2/1 ☐ b) Dekoder 2/4
☐ c) PLA sklop ☐ d) Multipleksor 3/1
☐ e) Dekoder 1/2

18. Arhitektura nekog sklopa koji ima ulaz cp te izlaze $q0$ i $q1$ (svi tipa `std_logic`) modelirana je VHDL-om prikazanim u nastavku. Sklop koristi T bistabil okidan padajućim bridom. O kojem se sklopu radi?

C
6.95

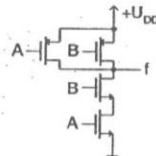
```

ARCHITECTURE some OF sklop IS
  COMPONENT tbistabil IS
    PORT (cp, t: IN std_logic, q, qn: OUT std_logic);
  END COMPONENT;
  SIGNAL qi: std_logic;
BEGIN
  b1: tbistabil PORT MAP (cp, '1', qi, open);
  b2: tbistabil PORT MAP (q1, '1', q1, open);
  q0 <= qi;
END some;

```

☐ a) Sinkrono binarno brojilo unaprijed ☐ b) Asinkrono binarno brojilo unatrag
☒ c) Sinkrono binarno brojilo unatrag ☐ d) Sinkroni registar
☐ e) Asinkrono binarno brojilo unaprijed

19. Neka funkcija ostvarena je CMOS tehnologijom prema slici. O kojoj se funkciji radi?



- ☐ a) I ☐ b) Ex-ILI
☐ c) ILI ☐ d) NILI
☒ e) NI

20. Korištenjem K-tablica pronaći minimalni zapis funkcije f u obliku sume parcijalnih produkata, ako je $f(A, B, C, D) = \sum m(5, 7, 12, 15) + \sum d(3, 10, 13)$. Rezultat minimizacije je:

- C
1 ☐ a) $A \cdot B + B \cdot D$ ☐ b) $\bar{C} \cdot D + B \cdot C \cdot D$
☒ c) $A \cdot B \cdot \bar{C} + B \cdot D$ ☐ d) $\bar{C} \cdot D + B \cdot D + A \cdot \bar{B} \cdot \bar{C}$
☐ e) $\bar{C} \cdot D + B \cdot D$

21. Distanca nekog koda iznosi 7. Koliko pogrešaka ovaj kod može ispraviti/otkriti?

- 2.95 ☐ a) 0/0 ☐ b) 6/6
☐ c) 3/5 ☒ d) 3/6
☐ e) 2/6

$d \geq 2t - 1$ $t = 6$
 $d \geq 2g - 1$ $g = 9$

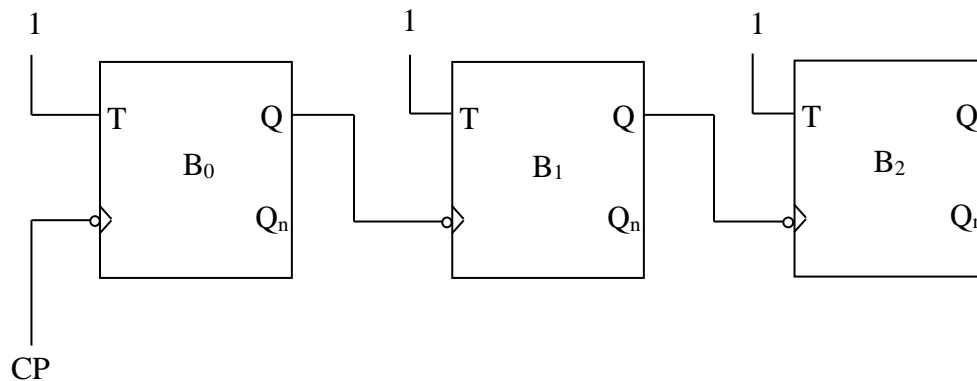
17. Ulazi su x, y i z, izlaz je o. Ovisno o z-u, na izlaz se proslijeđuje x ili y. A to je upravo funkcija mux-a. Z je adresni ulaz, x je podatkovni ulaz 0, a y podatkovni ulaz 1. Znači radi se o mux-u 2/1. Odgovor: a)

*Napomena (Mux 3/1 ne postoji, jer mux može imati samo 2^n podatkovnih ulaza, znači 4/1, 8/1...)

18. Binarno asinkrono brojilo može imati 4 kombinacije:

- a) padajućim bridom okidano brojilo unaprijed
- b) padajućim bridom okidano brojilo unatrag
- c) rastućim bridom okidano brojilo unaprijed
- d) rastućim bridom okidano brojilo unatrag

Svako od ovih brojila se razlikuje po načinu spajanja izlaza i pobudi na ulazu CP-a svakog bistabila. Mi smo radili samo brojila ostvarena T-bistabilima, tak da se za ostala ne trebate brinut. Dolje je nacrtan primjer asinkronog binarnog brojila unaprijed okidanog padajućim bridom CP-a:



Za početak treba primijetiti da su težine bistabila poredane obrnuto (bit B_0 je bit najmanje težine). Pa krenimo po redu... predpostavimo da je početno stanje svih bistabila 0 (znači svi Qovi su u 0). Na prvi padajući brid, Q_0 se mijenja u 1, promjena iz 0 u 1 prouzrokuje rastući brid, znači u bistabilu B_1 se ništa ne događa, kao ni u B_2 . Nakon prvog CP-a znači na izlazima dobivamo podatak 001 (gledamo s desna na lijevo zbog težina!!). Na sljedeći CP impuls, Q_0 pada na nulu, što je padajući brid, znači okine se bistabil B_1 i njegov Q_1 poraste na 1, što je pak rastući brid, tako da se u bistabilu B_2 još uvijek ne događa ništa. Nakon drugog impulsa CP-a na izlazima dobivamo podatak 010. Dalje 011, 100, i tak uglavnom binarno broji.

Ako su pak ovako spojeni bistabili *okidani rastućim bridom*, dobili bismo brojilo unatrag: za podatak 000, na prvi CP, Q_0 bi porasao u 1, što je rastući brid, zato bi i Q_1 porasao u 1, pa bi i on slao rastući brid bistabilu B_2 , koji bi također prešao u 1, te bi nam tako sklop na izlazu dao 111. Sljedeći rastući brid bi srušio Q_0 u 0, što bi poslalo padajući brid bistabilu B_1 , ali on takve bridove ne šljivi (čeka rastući), dok B_2 naravno također ne bi imao promjene, jer on nije nikakav brid dobio.

Sad ostavljam na vama da po ovakvoj logici probate sami shvatiti da ako na CP-ove bistabila okidanih padajućim bridom spojite Q_n prethodnog, dobit ćete asinkrono brojilo

unatrag, a za isto spojene bistabile okidane rastućim bridom, dobili bi asinkrono brojilo unaprijed. Ko to ne kuži (nakon što bar 5 puta pročita kaj sam sad napiso) nek ode plakat mami!

A sad (tek) idemo na zadatak...

Ove dvije naredbe kaj počinju sa b1: i b2: bi u c-u mi nazvali poziv funkcije t-bistabil, koja kao argumente uzima ovo kaj je u zagradi, a njezin prototip je tamo gore di piše component.

Znači, za bistabil1: na cp dovodi cp sklopa (znači ovo će biti bistabil najmanje težine), na t dovodi logičku 1 (naravno, kaxam i ja gore nacrtio), izlaz q pridružuje internoj varijabli qi, a izlaz qn ne koristi (zato piše open).

Bistabil2: na cp dovodi internu varijablu qi, koju je maloprije definio kao izlaz q prethodnog bistabila, tako da u biti na cp ovog bistabila dovodi izlaz q prethodnog, kako sam i ja to nacrtio gore. Na t-u je opet logička 1, a izlaz ovog bistabila pridružuje izlazu sklopa q1.

Nakon definiranja funkcija bistabila 1 i 2, pridružuje izlaz prvog bistabila izlazu sklopa q0.

Budući da je u zadatku navedeno da se bistabili okidaju padajućim bridom, a cp-i sukcesivnih bistabila su spojeni na izlaz q prethodnog, radi se o asinkronom brojilu unaprijed.

Odgovor: e)

19. Prije neg kaj se zaletimo, treba primijetiti da su u ovom sklopu dvije vrste cmosova: pmos i cmos (demosi na žalost nema ☹). Pmos je obično spojen na napajanje, osim ak vas Čupić oće grdo sjebati, pa makar ne radio sklop. Ali da budete sigurni, provjerite di je naznačena strelica: ako je na collectoru (na gornjoj žičobi) onda je pmos, ak je na emitteru (na donjoj žičobi) onda je nmos. Ako se dogodi da nema strelica, onda ćete pmos prepoznati po tome da ima kružić ispred gatea (ova srednja žičoba), što označuje da se varijable na pmosu gledaju komplementirano!! Daklem, kad su A i B 00, na pmosove gledamo kao da su im na gateovima 11 (jer komplementirane varijable gledamo kao što rekoh), znači oba pmosa vode, dok nmosovi ne vode, jer su oba gatea u 0, znači izlaz f je spojen na napon napajanja, a.k.a. u 1 je. Isto tako, za A i B 01 vodi pmos spojen na A, i ne vodi nmos spojen na A, tako da je opet napon spojen na f, i nije uzemljen- izlaz je opet 1, za 10 stvar je potpuno simetrična, izlaz je opet 1, a za AB=11, na pmos se dovode 0, znači nijedan od njih ne vodi, dok oba nmosa vode, što će reći da je izlaz f uzemljen, znači u 0 je. A kad pogledaš ovo zadovoljava uvjete funkcije NI.

Odgovor: e)

*Napomena: neke grupe su imale obrnut slučaj: gore su bili serijski pmosovi, a dole paralelno nmosovi, pa taj sklop daje NILI funkciju. A ak su vam ovak jednostavni sklopovi, onda mogu biti samo NI i NILI, jer su ostale funkcije puno kompliciranije za izvest mosfetima (sjetite se onog java appleta u 2. zadaći).

20. Da vam ne bi k-tablice objašnjavo! Al c) je točno.

21. Ako malo bolje pogleate, na skeniranom ispitu piše formula za ovo pitanje. No dobro. Formula za broj otkrivenih pogrešaka je $d \geq n+1$, a za broj ispravljenih je $d \geq 2n+1$, znači za distancu 7, broj otkrivenih pogrešaka će biti

$$7 \geq n+1$$

$$n \leq 6$$

A broj ispravljenih pogešaka je

$$7 \geq 2n+1$$

$$n \leq 3$$

Odgovor: d)

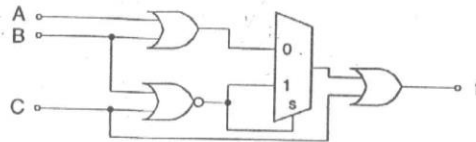
- 22) Funkcija od 4 varijable prikazana u kanonskom obliku produkta suma sadrži 11 maksterma. Ako istu funkciju prikažemo u kanonskom obliku sume produkata, koliko taj prikaz sadrži minterma?

a) 11
b) 10
c) 5
d) Ne može se odrediti
e) 16

- 23) U nekom digitalnom sustavu za pohranu cijelih brojeva koristi se binarni zapis u B-komplementu. Ako se za pohranu koristi 6 bitova, koji je najmanji broj koji se može prikazati?

a) -31
b) -64
c) -32
d) -16
e) 0

- 24) Koju funkciju obavlja sklop prikazan na slici?



a) $f(A, B, C) = \sum m(2, 4, 5, 7)$

b) $f(A, B, C) = \sum m(0, 3, 7)$

c) $f(A, B, C) = \sum m(0, 1, 2, 3, 4, 5, 6, 7)$

d) $f(A, B, C) = \sum m(0, 1, 2, 5, 7)$

e) $f(A, B, C) = \sum m(0, 1, 2, 5)$

- 25) Koji je od sljedećih minimalni potpuni sustav funkcija Booleove algebre?

a) {EX, ILI}

b) {ILI, NE}

c) {NE}

d) {NI}

e) {EX, NILI}

$$\overline{A \cdot B} = \overline{A} \cdot \overline{B}$$

14.1% 100%

54.8 95%

14

13 - 19

14.2

17.9

16.7

22. Ajoj, mintermi su ona mjesta di funkcija ima 1-ice, a makstermi ona di je funkcija u 0, tak da ak funkcija ima 4 varijable, znači da ima 16 kombinacija, od toga su 11 u 0 (zato kaj ima 11 maksterma), znači preostalih 5 je u 1-ici!

Odgovor: c)

23. Prednost 2-komplementa nad 1-komplementom je to što nema oznaku za negativnu 0! 0 se tretira kao pozitivan broj (bit predznaka je 0), a uz 6 bitova se s predznakom 0 (+) može prikazati 32 bita, znači od 0-31, a s predznakom 1 (-) se također može prikazati 32 broja, ali nula nije uključena, znači od -1 do -32 – najmanji broj je -32!

Odgovor: c)

24. Odma se vidi da je funkcija u 1 čim je $C=1$, jer se varijabla C direktno dovodi na ILI sklop. Najpametnije je odmah označavat za koje je kombinacije funkcija u 1:

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

Nakon toga, vidimo da je B nili C spojena i na podatkovni ulaz(1) muxa, i na adresni ulaz, što će reći da ona proslijeđuje samu sebe, tj. kad je ona u 1, proslijeđuje se 1, a taj 1 se proslijeđuje na ILI, znači funkcija je u 1 kad je B nili C u 1. Ako vam je teško računati s nili, pretvorite ga DeMorganom: $(B+C)' = B' \cdot C'$, pa idemo dodat to u prethodnu tablicu:

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

A kad $B' \cdot C'$ nisu u 1, onda se proslijeđuje $A+B$, pa provjerimo za preostale slučajeve kad su $A+B$ u 1:

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

1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Cijela tablica je popunjena, znači funkcija je suma svih minterma,
Odgovor: c)

25. Ako se sjetite prve zadaće, imali smo 1 zadatak gdje smo neku funkciju morali prikazati samo pomoću NI ili samo pomoću NILI funkcije – prema tome, i NILI i NI su potpuni sustavi Booleove algebre, a minimalni su, jer je to samo jedan operator (dobro ajde 2 elementarna – **ne** i **i/ili**). U ovom je zadatku kao odgovor od ta dva ponuđen samo NI.

Odgovor: d)

Courtesy of Zoltaaaaann!!