
lprs1

TEORIJA

1 BIT
0 → 0
1 → 1

2 BITA
00 → 0
01 → 1
10 → 2
11 → 3

3 BITA
000 → 0
001 → 1
010 → 2
011 → 3
100 → 4
101 → 5
110 → 6
111 → 7

4 BITA
0000 → 0
0001 → 1
0010 → 2
0011 → 3
0100 → 4
0101 → 5
0110 → 6
0111 → 7
1000 → 8
1001 → 9
1010 → 10 A
1011 → 11 B
1100 → 12 C
1101 → 13 D
1110 → 14 E
1111 → 15 F

OPŠTE PRAVILO

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

BINARNI → DECIMALNI

PARAN

10101**0** → 42
00110**0** → 12
01000**1** → 17

NEPARAN

DECIMALNI → BINARNI

20 → 0 1 0 1 0 0
(32 16 8 4 2 1)
13 → 0 0 1 1 0 1
4 → 0 0 0 1 0 0

HEXADEC → BINARNI

A7 → 1 0 1 0 0 1 1 1
D8 → 1 1 0 1 1 0 0 0
B6 → 1 0 1 1 0 1 1 0

BINARNI → HEXADEC

1010 0010 → 0xA2
1111 1110 → 0xFE
1100 0110 → 0xC6

KOMPLEMENT I, II

iA 10101011
KOMP1 01010100
KOMP2 + 1
01010101
kompl. 2

iA 10010111
KOMP1 01101000
KOMP2 01101001

iA 01010100
11
KOMP1 10101011
KOMP2 + 1
10101100

* 1 i 1 su 0 (1)

```
s_compl1 <= not(A);
s_compl2 <= not(A)+1;
```

10010.010 −13.75
01101.101
\+ 1
01101.110 = 13.75

FRAKCIONI BITOVI

500 250 125

. 0.5 0.25 0.125 ...

2^{-1} 2^{-2} 2^{-3}

$\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$

BINARNO → DECIMALNO

01011.110 → 11.75

10101.011 → 21.375

00111.111 → 7.875

DECIMALNO → BINARNO

9.25 → 101.010

11.375 → 1011.011

4.5 → 100.100

-13.75 → 01101.110 (u)

01101.110

−1

01101.101

10010.010

= Y

01101.110

10010.001

+ 1

10010.010

PAZITI NA OZNAČENE-NEOZNAČENE

- KOMPLEMENT 1 ILI KOMPLEMENT 2

BITNO DA GLEDAŠ DA LI JE U ZAD REKAO DA JE BROJ ORIGINAL ILI JE VEĆ NEKI OD KOMPLEMENATA!!

dobijemo tabelu

sign → OZNAČENI { 0 → + ; 1 → −

S	I I I I .	F F F	
0	0 1 0 1 .	1 1 0	+ 5.75
1	0 1 1 0 .	0 0 1	−
1	0 1 0 0 .	1 1 0	

1 0 1 . 1 1 0 => 5.75

NEGATIVAN: RADI SE KOMPLEMENT 2 !

10110.001 ← -9.875 (u)

1° 01001.110

\+ 1 ← DODAJE SE NA FRAKCIONI DEO !

01001.111 = 9.875 (u)

ORS 0 1 0 1 (+5)

K1 1 0 1 0

\+ 1

K2 1 0 1 1

-8+2+1 = -5

-3.25:

3.25 => 011,01

↑ ZNAK!

011.01

100.10

\+ 1

100.11

10100.110

01011.001

\+ 1

01011.010

-11.4

* 01101.101 = 13.625

NE KLIKAJ SAMO 12.625 NEGO -12.625

S I I I I . F F F

0 1 1 0 1 . 1 0 1 + 13.625

1 0 0 1 1 . 0 1 1 − 12.625

10011.011 =

1° 01100.100

\+ 1

01100.101 = 12.625

BULOVA ALGEBRA

KARNOOVE MAPE

$\vec{A} \rightarrow 0$

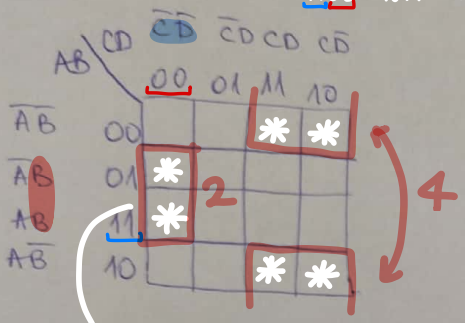
$A \rightarrow 1$

1° $f^1(ABCD) = (0\,0\,0\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,0\,0)$

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

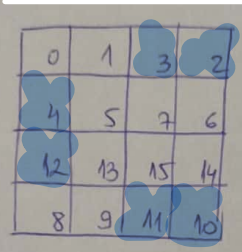
$f^1(ABCD) = (12, 11, 10, 4, 3, 2)$

1100 1011 1010 0100 0011 0010

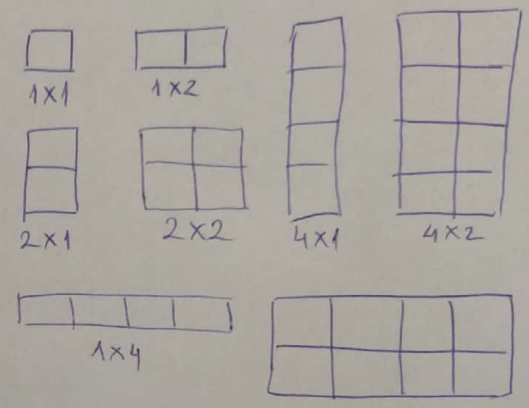


$F = B\bar{C}\bar{D} + \bar{B}C$

ILI NAPAMET:



KARTE:



2° $f^1(ABCD) = (0,5,7,13,15) = \bar{A}\bar{B}C\bar{D} + BD$ ← RADI SAM POSLE

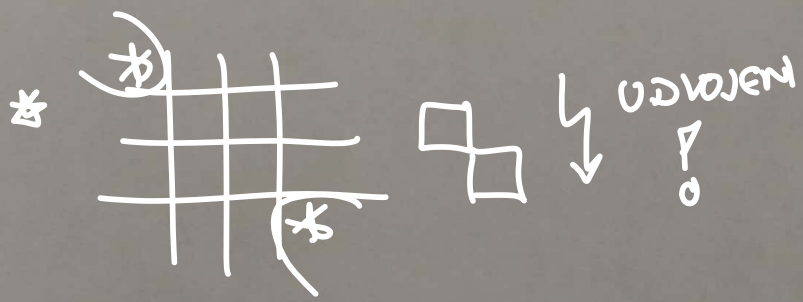
3° $f^1(ABCD) = (8,9,10,12,13,14)$

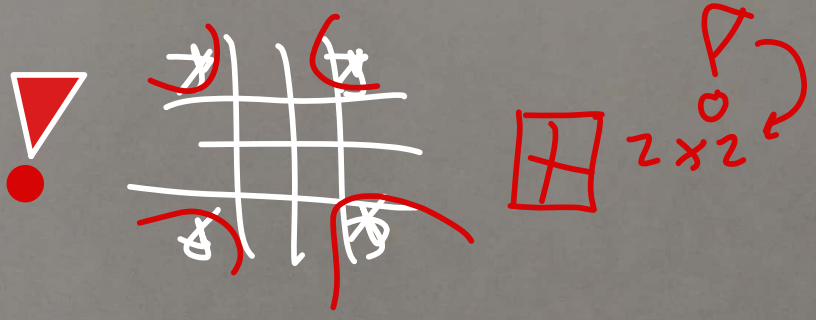
4° $f^1(ABCD) = (1111010000001010)$

5° $f^1(ABCD) = (0100011001000110)$

6° $f^1(ABCD) = (0,3,7,11,15)$

7° $f^1(ABCD) = (1010010110100101)$





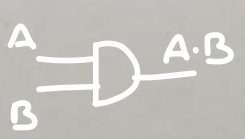
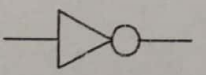
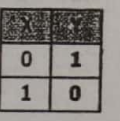
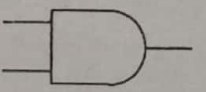
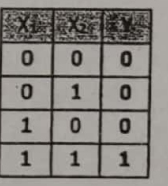
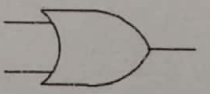
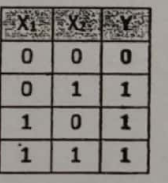
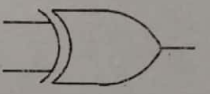
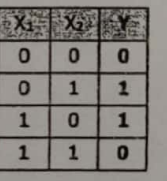
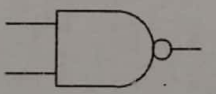
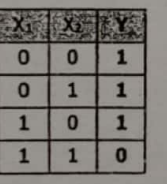
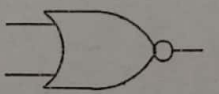
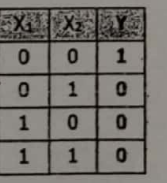
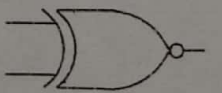
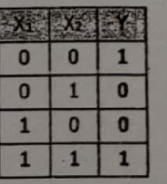
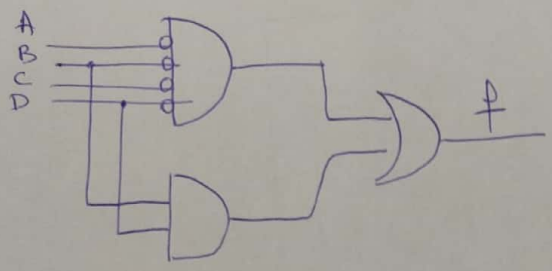


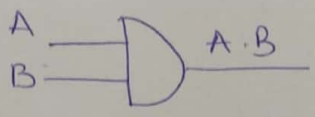
Tabela 1-1. Digitalna logička kola

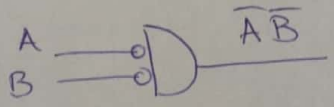
NOT		$Y = \bar{X}$	
AND		$Y = X_1 X_2$	
OR		$Y = X_1 + X_2$	
XOR		$Y = X_1 \oplus X_2$	
NAND		$Y = \overline{X_1 X_2}$	
NOR		$Y = \overline{X_1 + X_2}$	
XNOR		$Y = \overline{X_1 \oplus X_2}$	

4

1° $f = BD + \bar{A}\bar{B}\bar{C}\bar{D}$







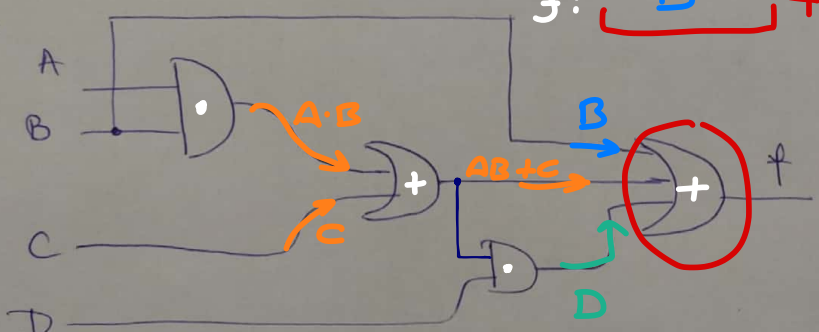
2° $f = A\bar{C} + AC\bar{D}$

3° $f = \bar{B}CD + C\bar{D}$

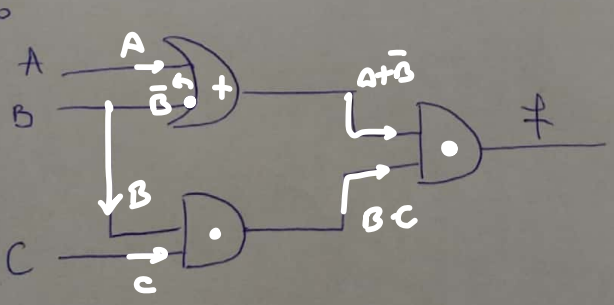
4° $f = AB + \bar{B}D$

5° $f = CD + \bar{A}\bar{B}CD$

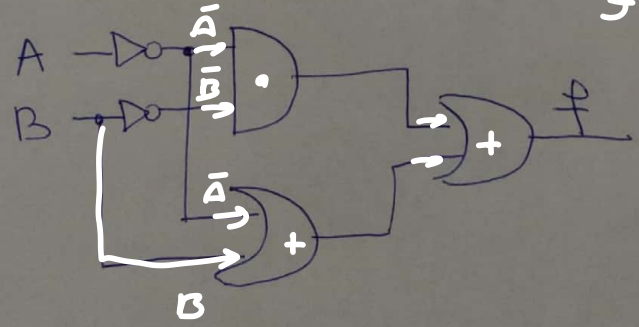
6° $f: \underbrace{B} + \underbrace{AB+C} + \underbrace{(AB+C)\cdot D}$



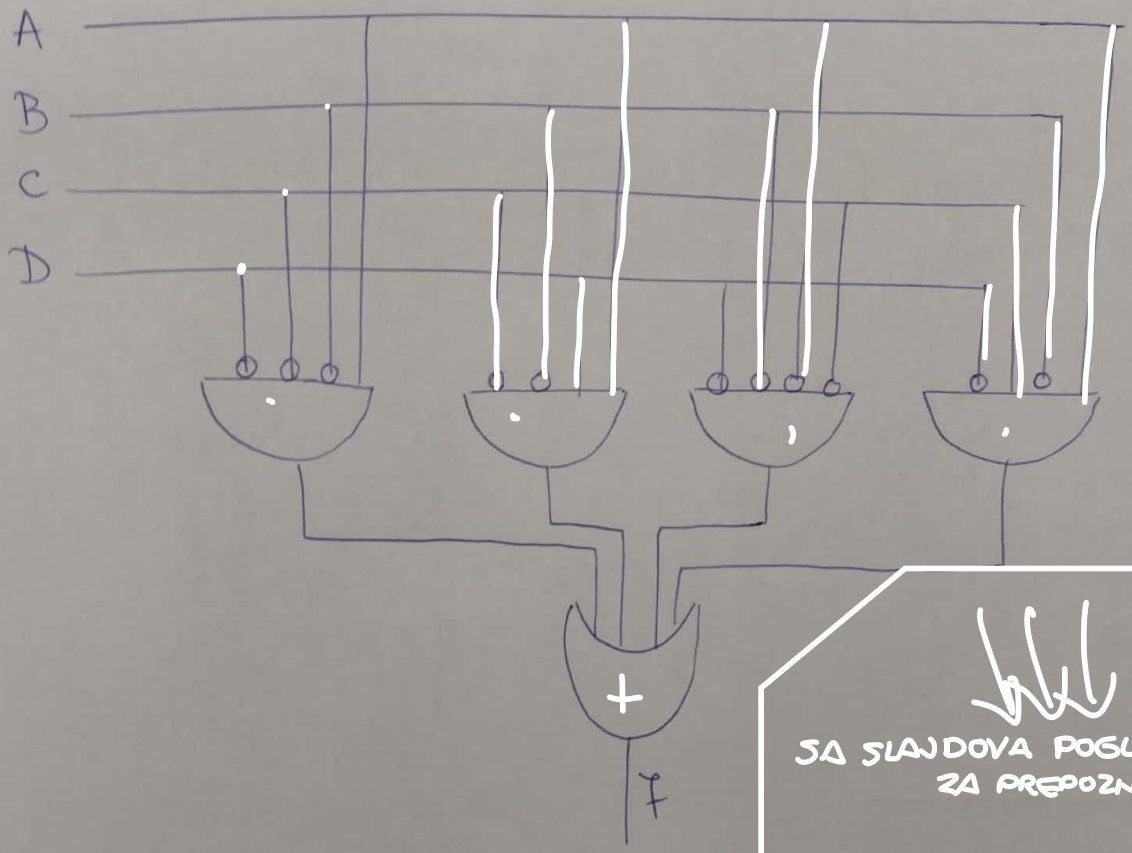
7° $f: \underbrace{A+\bar{B}} \cdot \underbrace{B\cdot C}$



8° $f: \underbrace{\bar{A}\cdot\bar{B}} + \underbrace{(\bar{A}+B)}$



9° SA KLK

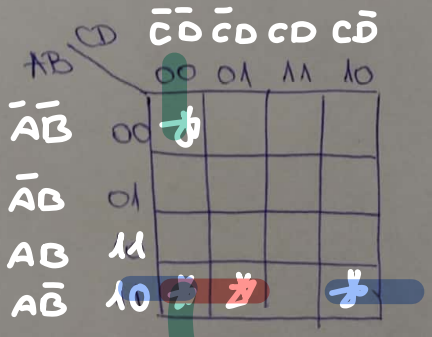


!!! !

SA SLAJDOVA POGLEDATI KODOVE ZA PREPOZNATI

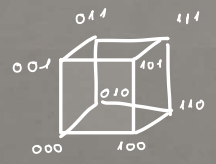
$$f = A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$$

$$f = A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$$



$$f = A\bar{B}\bar{C} + A\bar{B}\bar{D} + \bar{B}\bar{C}\bar{D}$$

SAM URADIO HAHAH



HEMINGOVA (KOCKA) RASTOJANJE

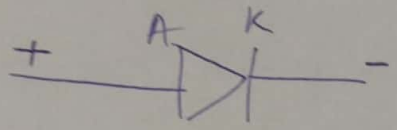
SPOJNICE HAHAHAH

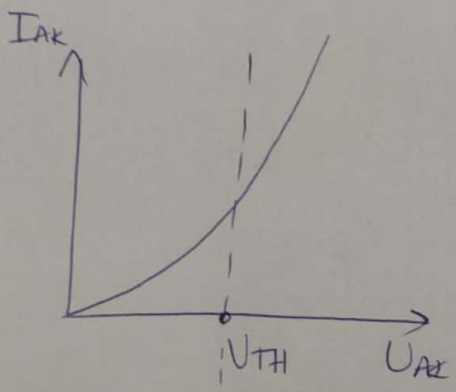
0	← NULA
1	← JEDINICA
L	← SLABA NULA
H	← SLABA JEDINICA
Z	← VISOKA IMPEDANSA
W	← SLABO NEPOZNATO (KOLIZIJA)
X	← NEPOZNATO (KOLIZIJA)
U	← NEINICIJALIZOVANA
–	← NEDEFINISANO

VISE ZA ISPIT

OSNOVNE PREKIDAČKE KOMPONENTE

DIODA

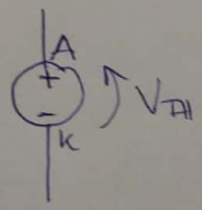




A → ANODA
K → KATODA

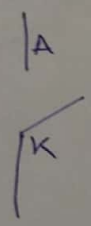
KRITERIJUM UKLJUČENJA DIODE:

1) $U_{AK} \geq V_{TH}$ → DIODA PROVODI



→EKVIVALENTA ŠEMA DIODE AKO PROVODI

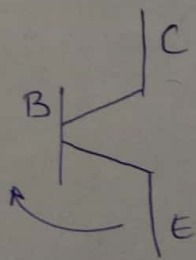
2) $U_{AK} < V_{TH}$ → DIODA NE PROVODI



→EKVIVALENTA ŠEMA DIODE AKO NE VODI

TRANZISTOR

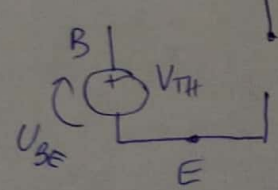
NPN TIP:



B → BAZA
E → EMITER
C → KOLEKTOR

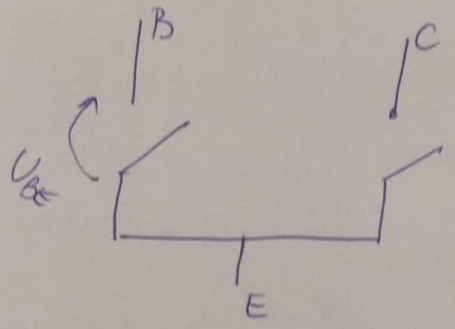
KRITERIJUM PROVOĐENJA TRANZISTORA

1) $U_{BE} \geq V_{TH}$ → TRANZISTOR PROVODI



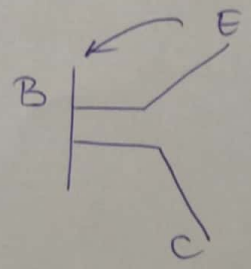
→EKVIVALENTA ŠEMA AKO TRANZISTOR RROVODI

2) $U_{BE} < V_{TH}$ → TRANZISTOR NE PROVODI



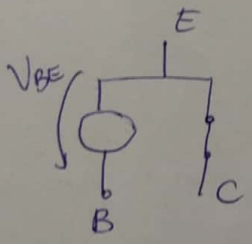
→ EKVIVALENTA ŠEMA KADA TRANZISTOR NE PROVODI

PNP TIP



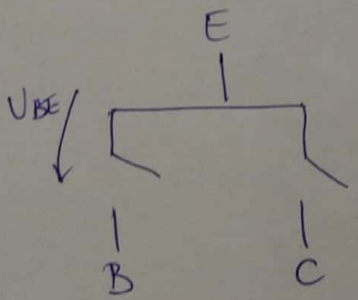
KRITERIJUM UKLJUČENJA

1) $U_{BE} \leq -|V_{TH}|$



→ EKVIVALENTA ŠEMA KADA TRANZISTOR PROVODI

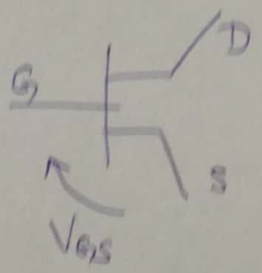
2) $U_{BE} > -|V_{TH}|$



→ EKVIVALENTA ŠEMA KADA TRANZISTOR NE PROVODI

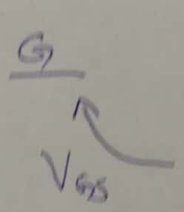
MOS FET TRANZISTOR

N-MOS



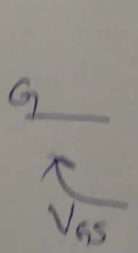
KRITERIJUM UKLJUČENJA TRANZISTORA

1) $V_{GS} \geq V_{THMOS}$ → TRANZISTOR PROVODI



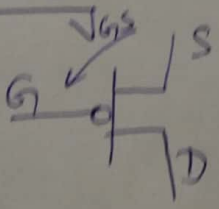
← EKVIVALENTA ŠEMA AKO TRANZISTOR PROVODI

2) $V_{GS} < V_{THMOS}$ → TRANZISTOR NE PROVODI



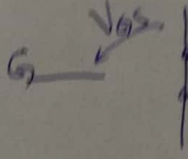
← EKVIVALENTA ŠEMA AKO TRANZISTOR NE PROVODI

P-MOS



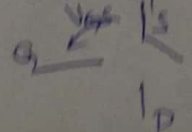
KRITERIJUM UKLJUČENJA TRANZISTORA

1) $V_{GS} \leq -|V_{THMOS}|$ → TRANZISTOR PROVODI



→ EKVIVALENTA ŠEMA AKO TRANZISTOR PROVODI

2) $V_{GS} > |V_{THMOS}|$

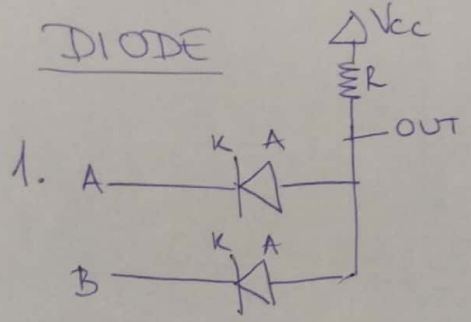


→ EKVIVALENTA ŠEMA AKO TRANZISTOR NE PROVODI

REALNA LOGIČKA KOLA

$0 < V_{TH} < V_{cc}$
↓
0,6V

DIODE



A – ANODA
K – KATODA

ISTINITOSNA TABLICA:

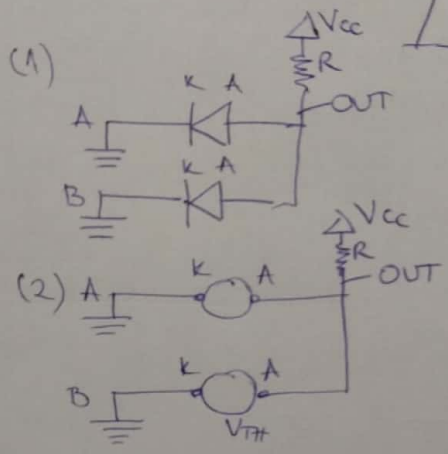
A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

"U" KOLO

"00": NP: OBE ISKLJUČENE: (UZEMLJIMO IH)

D_1: $U_A = V_{cc}$
$U_K = 0$
$U_{AK} = U_A - U_K = V_{cc} > V_{TH}$
PROVODI

D_2: $U_A = V_{cc}$
$U_K = 0$
$U_{AK} = U_A - U_K = V_{cc} > V_{TH}$
PROVODI

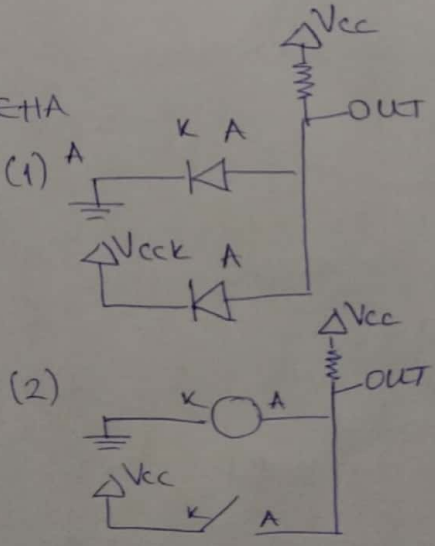


OUT = V_{TH} = "0"

"01" NP: D_1 ISKLJUČENA, D_2 UKLJUČENA

D_1: $U_A = V_{cc}$
$U_K = 0$
$U_{AK} = V_{cc} > V_{TH}$
PROVODI

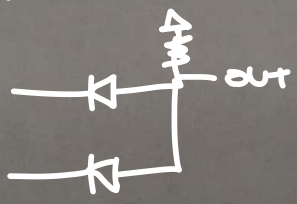
D_2: $U_A = V_{cc}$
$U_K = V_{cc}$
$U_{AK} = 0 < V_{TH}$
NE PROVODI



OUT = "0"

"01 = "10"

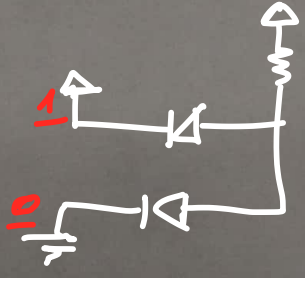
NA KLI2
10 VARIJACIJA



UPAMTIS SLIKU I TST

1) I
2) ILI
3) NI
4) NILI

2° TIP KOJI DA:



1 1
2) 0
3) U
4) –
5) Z

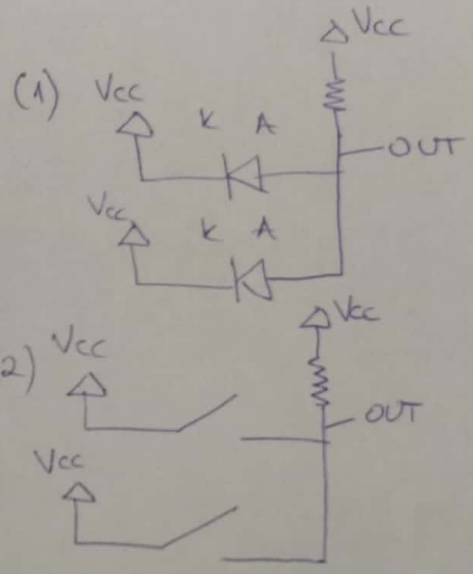
← sig nesto od ova dva 50/50

10

"11": ПП: ОБЕ ДИОДЕ ИЗКЛЮЧЕНИ

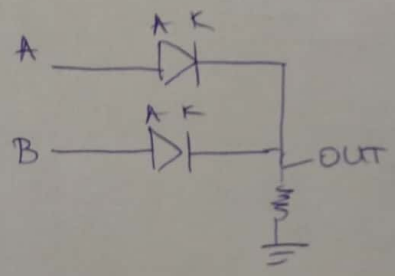
D_1: $U_A = V_{CC}$
$U_K = V_{CC}$
$U_{AK} = 0 < V_{TH}$
не проводи

D_2: $U_A = V_{CC}$
$U_K = V_{CC}$
$U_{AK} = 0V$
не проводи



$OUT = V_{CC} = 1$

2.



ISTINITOSNA TABLICA:

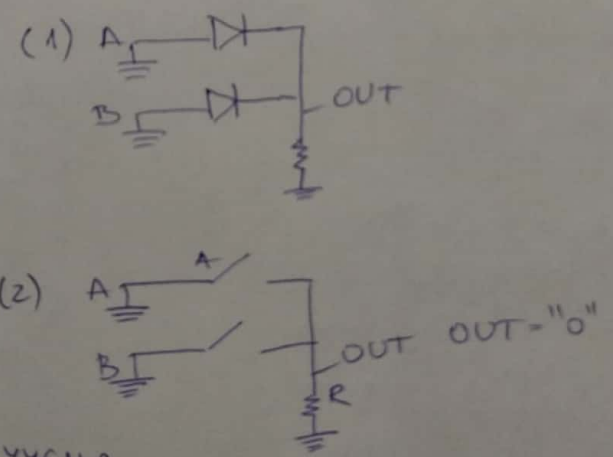
A	B	
0	0	0
0	1	1
1	0	1
1	1	1

"ИЛИ" КОЛО

"00": ПП: ОБЕ СЪ ИСКЛЮЧЕНИ

D_1: $U_A = 0$
$U_K = 0$
$U_{AK} = 0 < V_{TH}$
не проводи

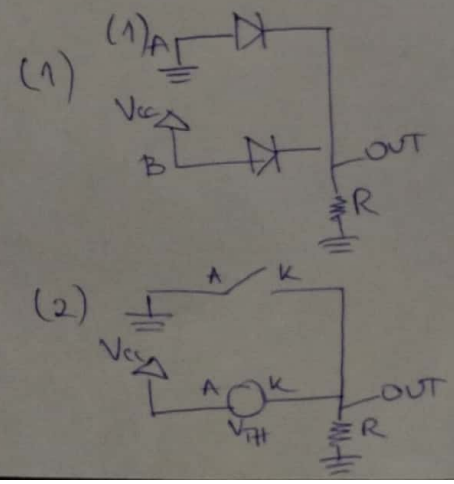
D_2: $U_A = 0$
$U_K = 0$
$U_{AK} = 0 < V_{TH}$
не проводи



"01": ПП: D_1 ИСКЛЮЧЕНА, D_2 ВКЛЮЧЕНА

D_1: $U_A = 0$
$U_K = 0$
$U_{AK} = 0 < V_{TH}$
НЕ ПРОВОДИ

D_2: $U_A = V_{CC}$
$U_K = 0$
$U_{AK} = V_{CC} > V_{TH}$
ПРОВОДИ



OUT = "1"

"01" = "10"

"11": ПП D_1 УКЪУЧЕНА, D_2 УКЪУЧЕНА

D_1: $U_A = V_{cc}$

$U_K = 0$

$U_{AK} = V_{cc} > V_{TH}$

ПРОВОДИ

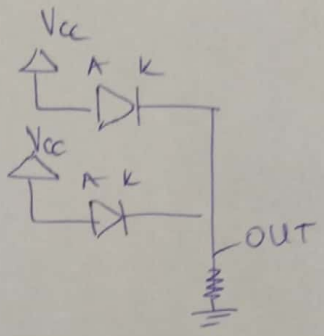
D_2: $U_A = V_{cc}$

$U_K = 0$

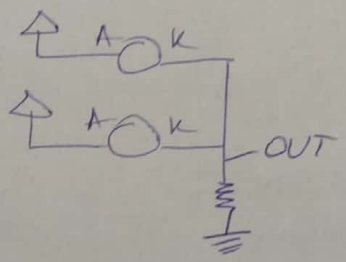
$U_{AK} = V_{cc} > V_{TH}$

ПРОВОДИ

(1)



(2)

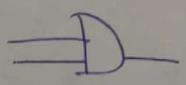


OUT = "1"

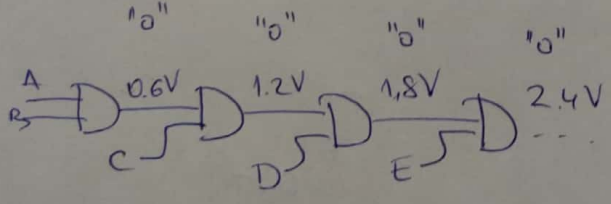
DRL ЛОГИКА – ПРОБЛЕМИ

– КАСКАДИРАНЕ!

"И" КОЛО

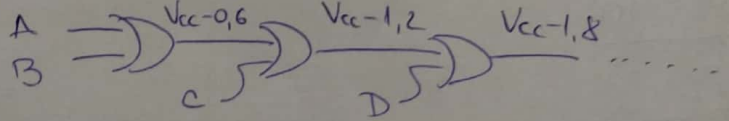


$$OUT = \begin{cases} "0" = 0{,}6V \\ "1" = V_{cc} \end{cases}$$

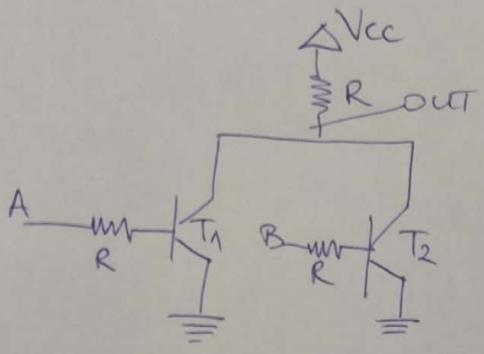


"ИЛИ" КОЛО

$$OUT = \begin{cases} "0" = 0V \\ "1" = V_{cc} - 0{,}6V \end{cases}$$



RTL - ОТПОРНИЧКО ТРАНЗИСТОРСКА ЛОГИКА



B – БАЗА
E – ЕМИТЕР

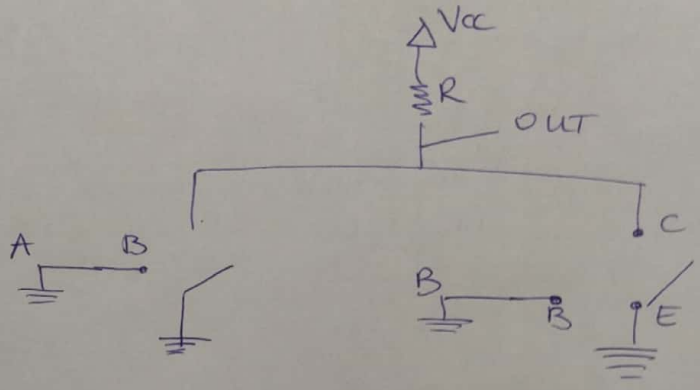
A	B	OUT
0	0	1
0	1	0
1	0	0
1	1	0

"НИЛИ" КОЛО

"00"

T_1: $U_B = 0$
$U_E = 0$
$U_{BE} = 0 < V_{TH}$
$T_1 \rightarrow$ ИСКЛ.

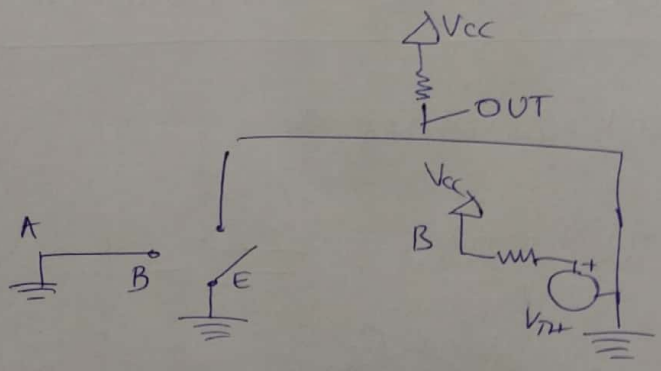
T_2: $U_B = 0$
$U_E = 0$
$U_{BE} = 0 < V_{TH}$
$T_2 \rightarrow$ ИСКЛ.



OUT = Vcc = '1'

"01": T_1: $U_B = 0$
$U_E = 0$
$U_{BE} = 0 < V_{TH}$
$T_1 \rightarrow$ ИСКЛ.

T_2: $U_B = V_{CC}$
$U_E = 0$
$U_{BE} = V_{CC} > V_{TH}$
$T_2 \rightarrow$ ВКЛ.



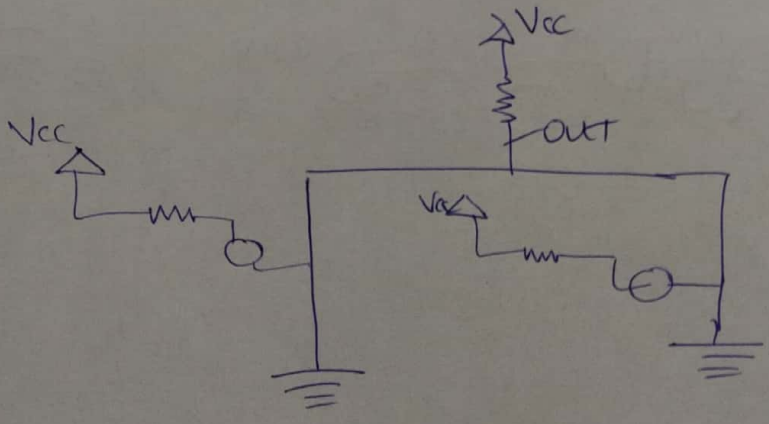
OUT = "0"

"01" = "10"

"11"

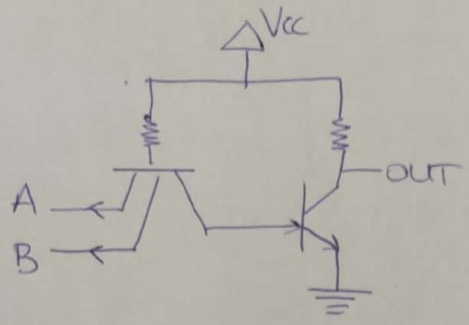
T_1: $U_B = V_{CC}$
$U_E = 0$
$U_{BE} = V_{CC} > V_{TH}$
T_1 ВКЛ.

T_2: $U_B = V_{CC}$
$U_E = 0$
$U_{BE} = V_{CC} > V_{TH}$
T_2 ВКЛУЧЕН



OUT = "0"

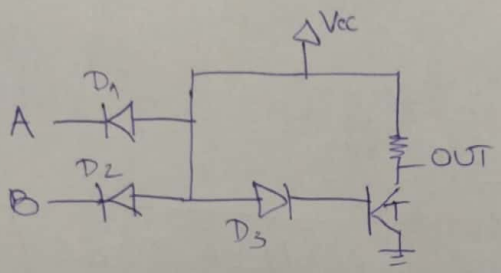
TTL - TRANZISTORSKO-TRANZISTORSKA LOGIKA



ISTINITOSNA TABLICA

A	B	OUT
0	0	1
0	1	1
1	0	1
1	1	0

"НИ" КОЛО



"00"

D_1: $U_K = 0V$ $U_A = V_{cc}$

$U_{AK} = V_{cc}$

ПРОВОДИ

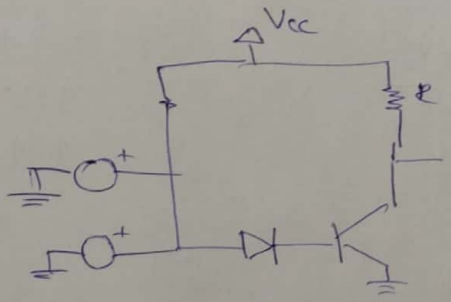
D_2: $U_K = 0V$ $U_A = V_{cc}$

$U_{AK} = V_{cc}$

ПРОВОДИ

D_3: $U_K = 0V$ $U_A = V_{cc}$

$U_{AK} = V_{cc}$



T: $U_B = V_{cc}$

$U_E = 0$

$U_{BE} > V_{TH}$

U_{BE} ВОДИ

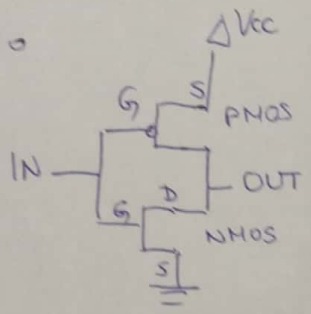
0

0

0

CMOS KOLA (NE, NILI, NI)

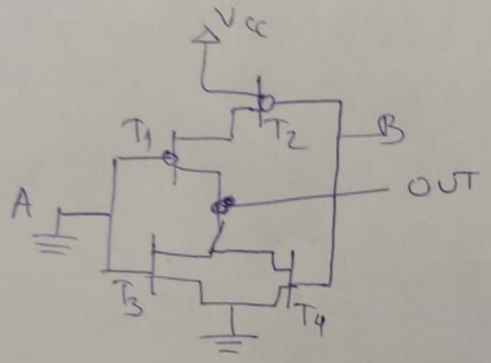
1°



IN	OUT
0	Vcc = 1
1	0

NE KOLO

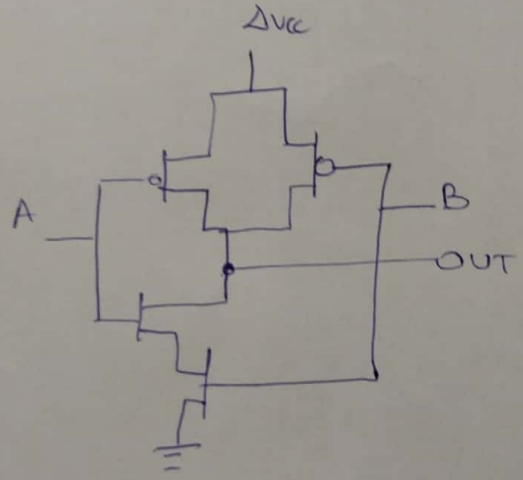
2°



A	B	OUT
0	0	Vcc = 1
0	1	0
1	0	0
1	1	0

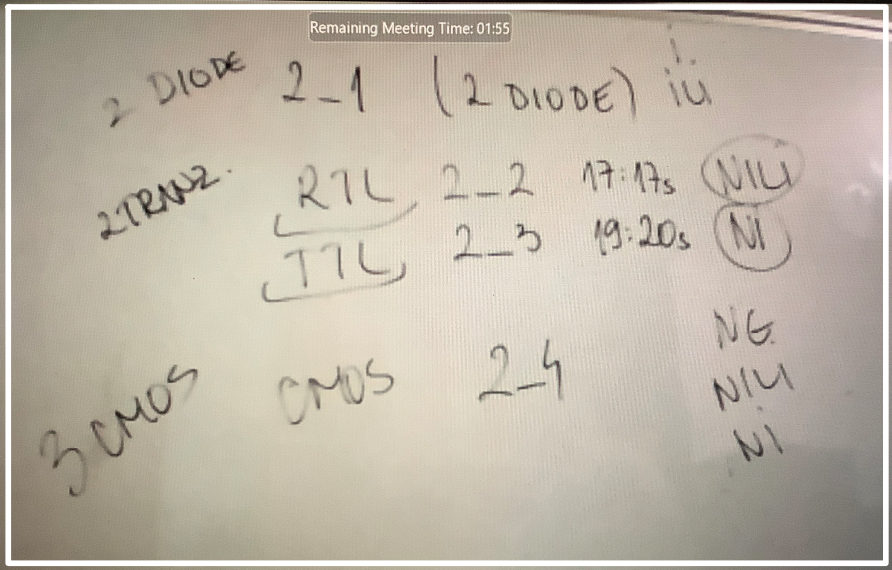
NILI KOLO

3°



A	B	OUT
0	0	Vcc = 1
0	1	Vcc = 1
1	0	Vcc = 1
1	1	0

NI KOLO



KOLO KOJA TREBA DA SE ZNAJU

ZADACI ZA VEŽBU:

S I I I I . F F F

a) 01010.110 = 10.75 ✓

b) 10110.011 = -9.625 ✓

01001.100 + 1 = 01001.101 = 9.625

c) 00101.001 = 5.125 ✓

d) 11001.100 = -6.5 ✓

00110.011 + 1 = 00110.100 = 6.5

e) 00001.111 = 1.875 ✓

f) 11110.010 = -1.75 ✓

00001.101 + 1 = 00001.110 = 1.75

g) 00111.011 = 7,375 ✓

h) 11001.101 = -6.375 ✓

00110.010 + 1 = 00110.011 = 6.375

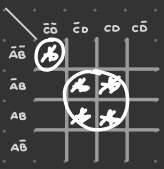
i) 00101.010 = 5.25 ✓

j) 10110.011 = -9.625 ✓

01001.100 + 1 = 01001.101 = 9.625

k) 01111.111 = 15.875 ✓

2°

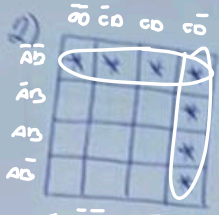


$f = BD + \bar{A}\bar{B}\bar{C}\bar{D}$ ✓

1)

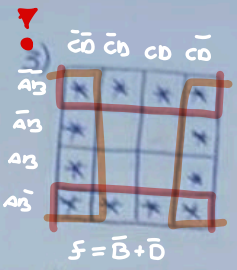
$f = \bar{B}\bar{D}$

2)



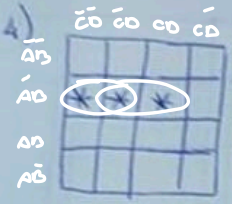
$f = \bar{A}\bar{B} + C\bar{D}$

3)



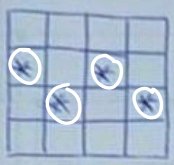
$f = \bar{B} + \bar{D}$

4)



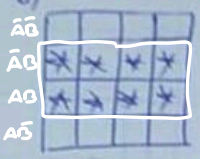
$f = \bar{A}B\bar{C} + \bar{A}BD$

5)



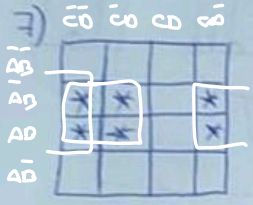
$f = \bar{A}B\bar{C}\bar{D} + AB\bar{C}D + \bar{A}BCD + AB\bar{C}\bar{D}$

6)



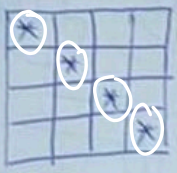
$f = B$

7)



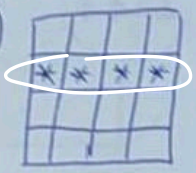
$f = B\bar{C} + B\bar{D}$

8)



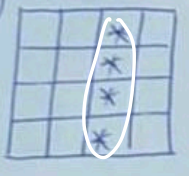
$f = \bar{A}B\bar{C}\bar{D} + \bar{A}BC\bar{D} + ABCD + AB\bar{C}\bar{D}$

9)



$f = \bar{A}B$

10)

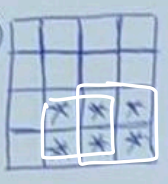


$f = CD$

11)

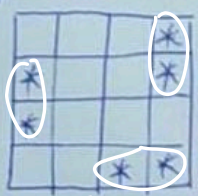
$f = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}D + \bar{A}DC$

12)



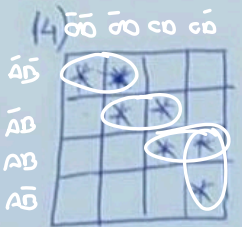
$f = AD + AC$

13)



$f = B\bar{C}\bar{D} + \bar{A}C\bar{D} + A\bar{B}C$

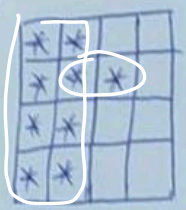
14)



$f = \bar{A}\bar{B}\bar{C} + \bar{A}BD + ABC + AC\bar{D}$

15)

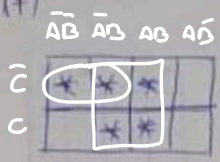
$f = \bar{C} + \bar{A}BD$



16) ?!

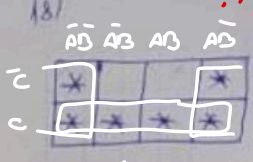
?! $f = \bar{A}C + \bar{A}B$

17) ?!

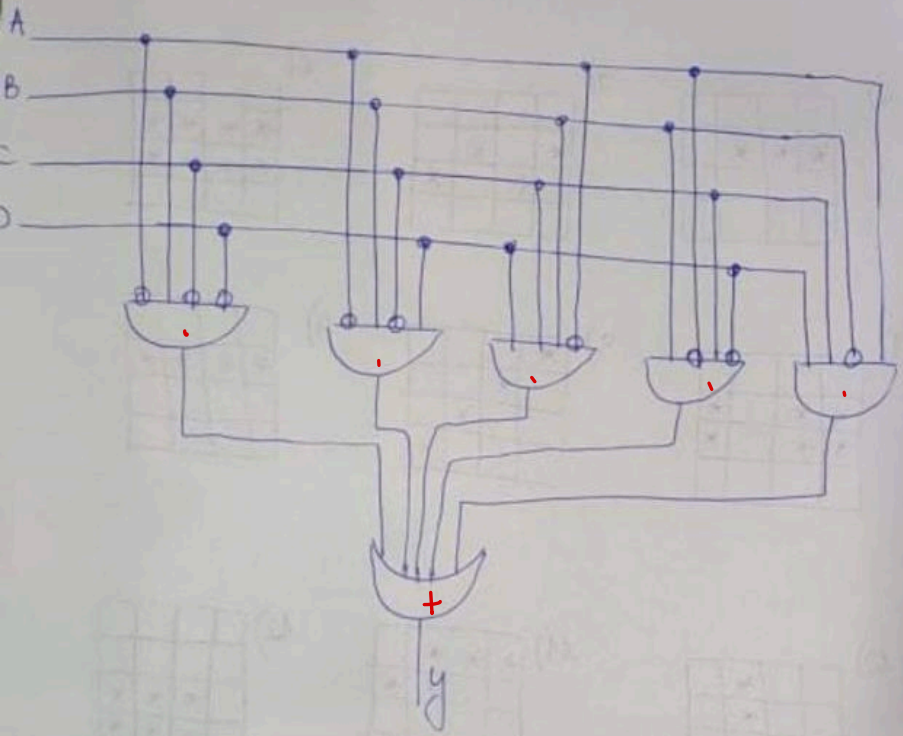


?! $f = \bar{A}\bar{C} + B$

18) ?!

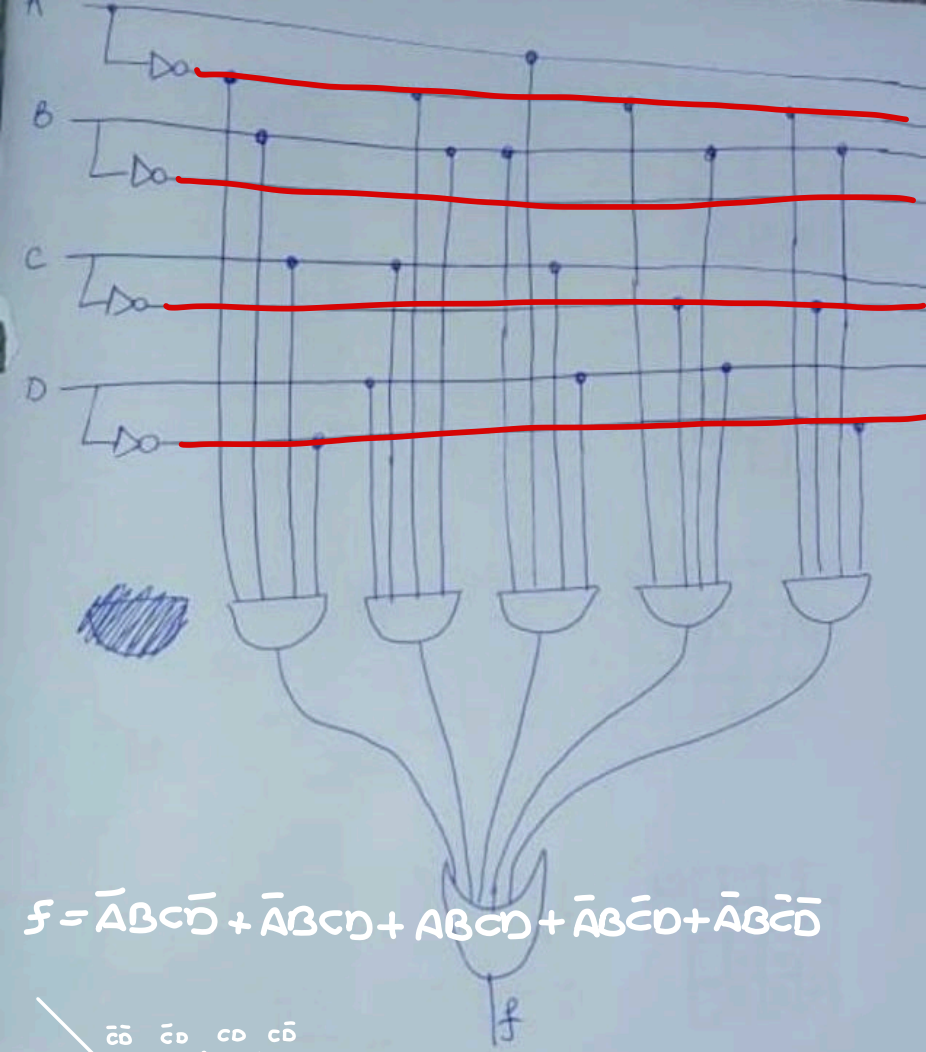


$f = \bar{B} + C$

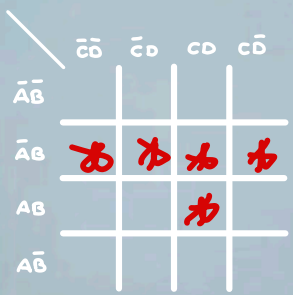


$f = \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BCD + \bar{A}BC\bar{D} + A\bar{B}CD$

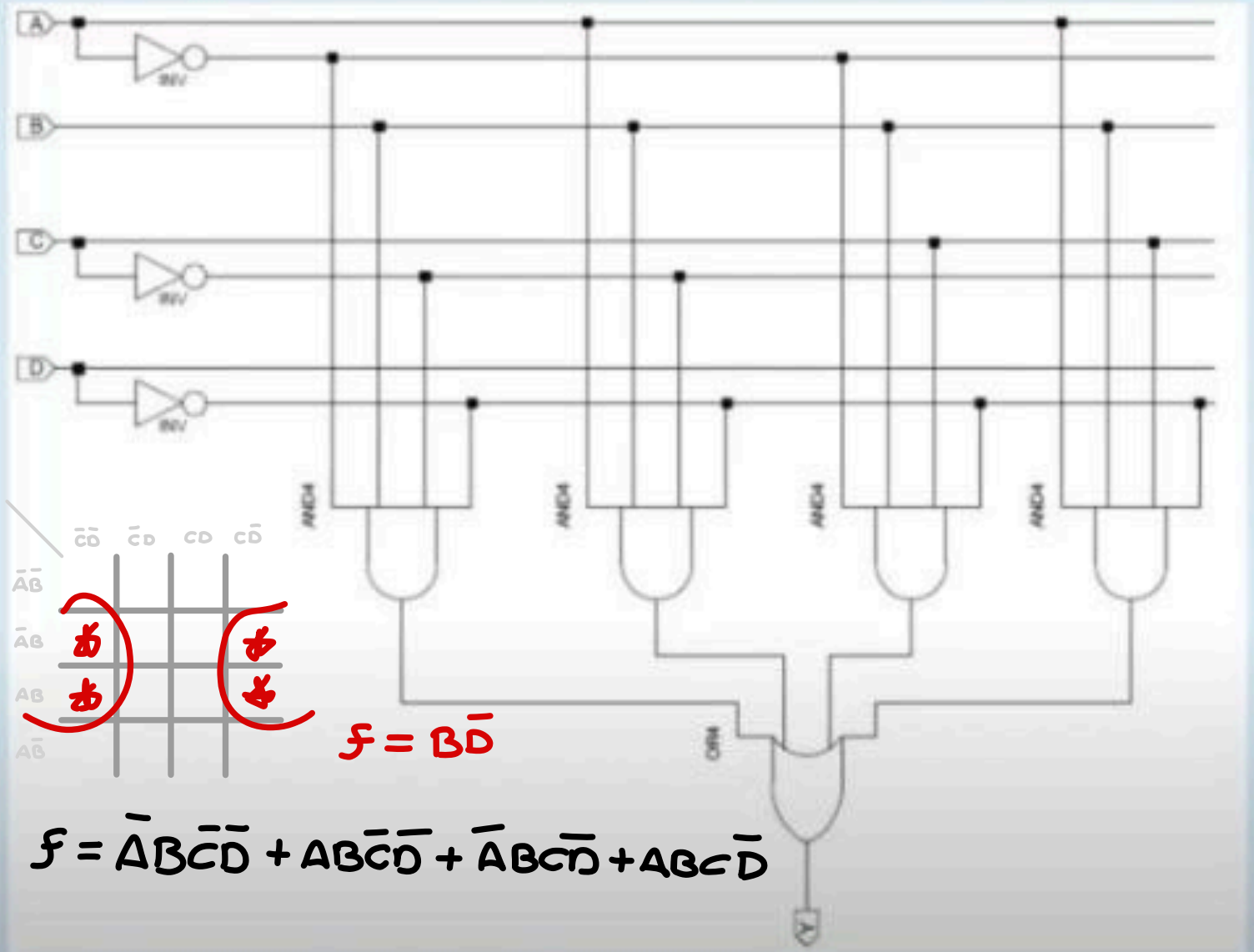
$f = \bar{A}B + A\bar{B}CD$



$$f = \bar{A}B\bar{C}\bar{D} + \bar{A}BCD + ABCD + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D}$$



$$f = \bar{A}B + BCD$$


A
B
C
D
AND4
AND4
AND4
AND4
OR4
F
$\bar{C}\bar{D}$ $\bar{C}D$ CD $C\bar{D}$
$\bar{A}\bar{B}$
$\bar{A}B$
AB
$A\bar{B}$
$F = B\bar{D}$
$F = \bar{A}B\bar{C}\bar{D} + AB\bar{C}\bar{D} + \bar{A}BC\bar{D} + ABC\bar{D}$

POKAZNA VEŽBA 3

Priprema za I kolokvijum

ZADATAK

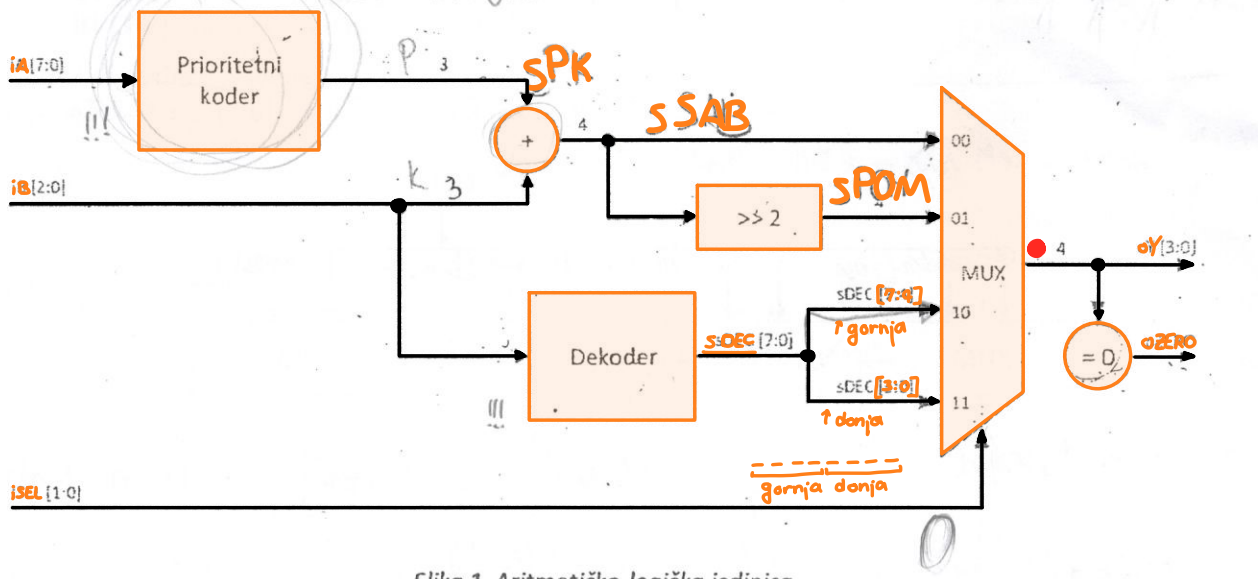
U VHDL jeziku za opis hardvera opisati i simulirati digitalni sistem prikazan na slici koji predstavlja jednostavnu aritmetičko-logičku jedinicu.

Ulazi digitalnog sistema:

- **iA [7:0]** – prvi ulazni operand koji se preuzima „dekodovan", tj. kao indeks jednog od 8 bita,
- **iB [2:0]** – drugi ulazni operand koji se preuzima „kodovan" u binarnoj predstavi,
- **iSEL [1:0]** – izbor operacije.

Izlazi digitalnog sistema:

- **oY [3:0]** – rezultat operacije,
- **oZERO** – oznaka da li je rezultat operacije jednak nuli.



Slika 1. Aritmetičko-logička jedinica

Operacije koje aritmetičko-logička jedinica podržava su:

- zbir dva ulazna operanda,
- zbir dva ulazna operanda podeljen sa 4, tj. aritmetički pomeren u desno za 2 mesta,
- gornja 4 bita dekodovane vrednosti drugog operanda,
- donja 4 bita dekodovane vrednosti drugog operanda.

Sistem simulirati izvršavajući svaku od operacija bar jednom, sa bar 2 različite vrednosti na svakom operandu. Bar jedan od slučajeva treba da rezultuje nulom i bar jedan od slučajeva brojem različitim od nule.

LOGIČKO PROJEKTOVANJE RAČUNARSKIH SISTEMA 1
Prvi kolokvijum – Grupa 3
13 Jan 2016

NAPOMENA

Za potrebe kolokvijuma koristiti direktorijum **C:\materija\LPRS1_X_Y_Z\KOL1**, gde je X oznaka studijskog programa (RA, E3), Y broj indeksa i Z godina upisa. Rešenje zadatka treba da se nalazi u tom direktorijumu.

ZADATAK

U VHDL jeziku za opis hardvera opisati i simulirati digitalni sistem prikazan na slici koji predstavlja jednostavnu aritmetičko-logičku jedinicu.

Ulazi digitalnog sistema:

- **iA [7:0]** – prvi ulazni operand,
- **iB [7:0]** – drugi ulazni operand,
- **iSEL [7:0]** – izbor operacije.

Izlazi digitalnog sistema:

- **oY [7:0]** – rezultat operacije.

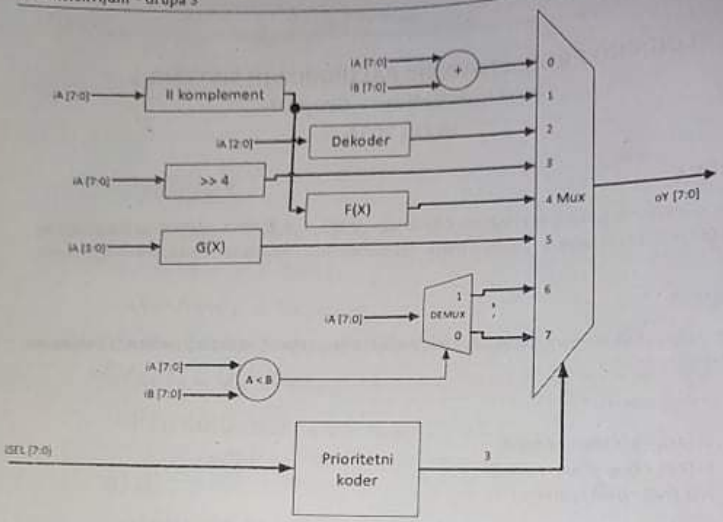
Operacije koje aritmetičko-logička jedinica podržava su:

- zbir dva ulazna operanda,
- komplement prvog ulaznog operanda u II predstavi,
- dekodovanje donja 3 bita prvog operanda,
- aritmetičko pomeranje prvog operanda za 4 mesta u desno,
- operacija F(X) = 3X,
- nepoznata operacija G(X) nad donja 4 bita prvog operanda (tabela 1),
- prvi operand, ako je A < B, u suprotnom 0,
- prvi operand, ako je A > B, u suprotnom A.

Izbor operacije se vrši preko prioritetnog kodera kome je prioritetniji bit sa većim indeksom.

Sistem simulirati kroz sledeće testne slučajeve (najmanje 10):

- na izlazu treba da bude 0x07 kao rezultat operacije G(X),
- na izlazu treba da bude 0x08 kao rezultat operacije dekodovanje,
- na izlazu treba da bude 0xCD kao rezultat operacije prvi operand, ako je A<B,
- na izlazu treba da bude 0xE8 kao rezultat operacije F(X),
- bar 6 slučajeva ostalih operacija, s tim da se svaka preostala operacija proba bar jednom.



Slika 1. Aritmetičko-logička jedinica

Tabela 1. Funkcija G(X): A[3:0] --> B[7:0]

A3	A2	A1	A0	B7-B4	B3	B2	B1	B0
0	0	0	0	0000	0	1	0	1
0	0	0	1	0000	0	1	1	0
0	0	1	0	0000	0	1	1	1
0	0	1	1	0000	1	0	0	0
0	1	0	0	0000	1	0	0	1
0	1	0	1	0000	1	0	1	0
0	1	1	0	0000	1	0	1	1
0	1	1	1	0000	1	1	0	0
1	0	0	0	0000	1	1	0	1
1	0	0	1	0000	1	1	1	0
1	0	1	0	0000	1	1	1	1
1	0	1	1	0000	0	0	0	0
1	1	0	0	0000	0	0	0	1
1	1	0	1	0000	0	0	1	0
1	1	1	0	0000	0	0	1	1
1	1	1	1	0000	0	1	0	0

TEORIJA ISP

	S1	S2	S3	S4	S5	S6	S7	S8	S9
0	S6/0	S6/0	S8/0	S8/1	S6/0	S9/1	S2/0	S3/1	S3/0
1	S6/1	S8/0	S8/1	S1/0	S8/1	S3/0	S3/1	S9/0	S6/1
	*	+	*	−	*	−	*	−	*

gde sve ima gore 0, dole 1

gde nam je 0, 0

gde nam je 1, 0

S8/1
IDE U ⌊ ⌊ IZLAZ
STANJE
(8)

NEMA S1 !!!

S1 S2 nisu istog stanja => X

S1 S3 jesu istog stanja:

S1	S3
S6/0	S8/0
S6/1	S8/1

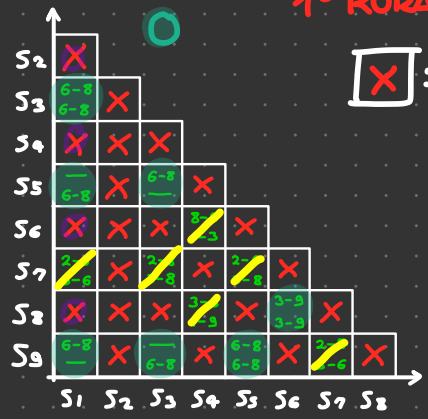
da imamo 8-8 umesto 6-8 onda pišemo u taj red samo "—"

	S1	S2	S3	S4	S5	S6	S7	S8
S2	X							
S3	6-8 6-8	X						
S4	X	X	X					
S5	— 6-8	X	6-8 —	X				
S6	X	X	X	8-9 1-3	X			
S7	2-6 3-6	X	2-8 3-8	X	2-6 3-8	X		
S8	X	X	X	3-8 1-9	X	3-9 3-9	X	
S9	6-8 —	X	— 6-8	X	6-8 6-8	X	2-8 3-6	X

za S1 S7 imamo 6-2, 6-3 ALI uvek pišemo, pišemo prvo MANJI BROJ: (2-6, 3-6)

NEMA S9 !!!

znamo da je samo S2 znaka ⊕
znači ni sa čim se neće slagati
pa možemo odmah x na sve!



1° KORAK treba da zapišemo svako x polje

1-2, 1-4, 1-6, 1-8
2-3, 2-4, 2-5, 2-6, 2-7, 2-8
3-4, 3-6, 3-8
4-5, 4-7, 4-9
5-6, 5-8
6-7, 6-9
7-8,
8-9

2° SADA od tih ↑ polja gledamo da li u zelenim poljima piše neka od tih komb ; onda ih precrtamo (dovoljno je da 1 od 2 broja bude na listi)
Zapisujemo koja su polja precrtana:

1-7, 3-7, 4-6, 4-8, 5-7, 7-9

+ ° SADA od tih ↑ polja gledamo da li je u nepreškrabanim ostao neki iz liste ŽUTIH, u ovom slučaju NIJE

da ima onda iterativno radimo do kraja

3° zaokružujemo ovo što je ostalo

4° ČITAMO AUTOMAT:

1. SVAKO stanje mora biti bar jednom
2. SVAKO stanje mora biti samo jednom

recimo

a) $(\underset{A1}{S_1}, S_3), (\underset{A2}{S_1}, S_5), (\underset{A7}{S_1}, S_9), (\underset{A4}{S_3}, S_5), (\underset{A5}{S_3}, S_9), (\underset{A6}{S_5}, S_9), (\underset{A7}{S_6}, S_8)$

NEMA S_2! ponavljaju se!

b) $(S_1, S_3, S_5, S_9)(S_6, S_8)(S_2)(S_4)(S_7)$

A1 A2 A3 A4 A5

TAČAN ODG

znači ugl po ova dva kriterijuma

ELIMINISATI NETAČNE

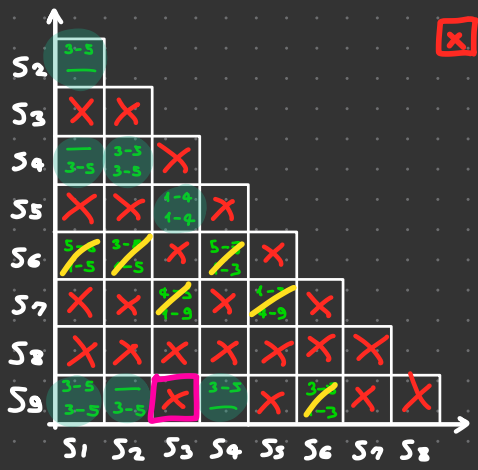
* bonus: KOLIKO U ZAD IMA AUTOMATA?

a 2 automata
b 5 automata
c 3 automata

} na osnovu rešenja hahahah

#

	S1	S2	S3	S4	S5	S6	S7	S8	S9
0	S3,0	S3,0	S9,1	S5,0	S1,1	S8,0	S3,1	S3,0	S3,0
1	S3,1	S5,1	S1,0	S3,1	S4,0	S1,1	S9,0	S5,0	S3,1
	*	*	-	*	-	*	-	0 = X svuda	*



X 1-3, 1-5, 1-7, 1-8,
2-3, 2-5, 2-7, 2-8,
3-4, 3-6, 3-8, 3-9,
4-5, 4-7, 4-8,
5-6, 5-8, 5-9,
6-7, 6-8,
7-8, 7-9,
8-9

□ 1-6, 2-6, 3-7, 4-6, 5-7, 6-9

da ovde ima tipa 3-5 izbacili bismo
i S_1-S_9, S_1-S_2, S_1-S_4, itd...

rešenje: $(S_1, S_2, S_4, S_9)(S_3, S_5)(S_6)(S_7)(S_8)$

da je ovo polje bilo zaokruženo morali bismo u
prvu zagradu ubaciti S3 (zbog S9) ali bi to
poruklo i S5 pa => $(S_1, S_2, S_3, S_4, S_5, S_9)(S_6)(S_7)(S_8)$

⇒ IMA 5 automata

1.

x | S	S1	S2	S3	S4	S5	S6	S7	S8	S9
0	$S_{2/0}$	$S_{6/0}$	$S_{8/0}$	$S_{9/1}$	$S_{6/0}$	$S_{3/1}$	$S_{2/0}$	$S_{3/1}$	$S_{8/0}$
1	$S_{8/1}$	$S_{8/0}$	$S_{8/1}$	$S_{1/0}$	$S_{8/0}$	$S_{1/0}$	$S_{5/1}$	$S_{8/0}$	$S_{6/1}$

REŠENJE: $(S_1, S_3, S_5, S_9)(S_6, S_8)(S_7)(S_2)(S_4)$

2.

x | S	S1	S2	S3	S4	S5	S6	S7	S8	S9
0	$S_{5/0}$	$S_{3/0}$	$S_{9/1}$	$S_{5/0}$	$S_{1/1}$	$S_{8/0}$	$S_{6/1}$	$S_{3/0}$	$S_{3/0}$
1	$S_{5/1}$	$S_{5/1}$	$S_{1/0}$	$S_{3/1}$	$S_{1/0}$	$S_{1/1}$	$S_{3/0}$	$S_{5/0}$	$S_{3/1}$

REŠENJE: $(S_1, S_2, S_4, S_9)(S_8)(S_7)(S_5)(S_3, S_6)$

3.

	S1	S2	S3	S4	S5	S6	S7	S8
W1	$S_{2/2_0}$	$S_{1/2_1}$	$S_{2/2_0}$	$S_{3/2_1}$	$S_{2/2_0}$	$S_{8/2_1}$	$S_{2/2_0}$	$S_{4/2_0}$
W2	$S_{2/2_1}$	$S_{4/2_0}$	$S_{2/2_1}$	$S_{2/2_0}$	$S_{4/2_1}$	$S_{4/2_0}$	$S_{2/2_1}$	$S_{4/2_1}$
W3	$S_{5/2_1}$	$S_{1/2_0}$	$S_{5/2_1}$	$S_{2/2_0}$	$S_{3/2_1}$	$S_{2/2_0}$	$S_{8/2_1}$	$S_{3/2_1}$

+ − + − + − + − +

REŠENJE: $(S_1, S_3, S_8)(S_2, S_4)(S_5, S_7)(S_6)$

4

	A_1	A_2	A_3	A_4	A_5	A_6	A_7
0	$A_{1/0}$	$A_{1/0}$	$A_{6/1}$	$A_{5/0}$	$A_{5/0}$	$A_{5/0}$	$A_{3/0}$
1	$A_{4/0}$	$A_{4/0}$	$A_{4/0}$	$A_{7/0}$	$A_{2/0}$	$A_{4/0}$	$A_{4/0}$

REŠENJE: $(S_1, S_5, S_6)(S_2, S_4)(S_3)(S_7)$

+ PRED.

5.

x | S	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
0	$S_{2/0}$	$S_{4/0}$	$S_{6/0}$	$S_{8/0}$	$S_{10/0}$	$S_{4/0}$	$S_{10/0}$	$S_{8/0}$	$S_{10/1}$	$S_{4/0}$	$S_{2/0}$	$S_{2/0}$
1	$S_{3/0}$	$S_{5/0}$	$S_{7/0}$	$S_{9/0}$	$S_{9/0}$	$S_{12/0}$	$S_{12/0}$	$S_{11/0}$	$S_{11/0}$	$S_{11/0}$	$S_{11/0}$	$S_{11/0}$

Sub Rz, Rx, Ry

1) Rz ← Rx - Ry TAČAN
2) Rx ← Ry - Rz
3) Ry ← Rz - Rx

OVO BUDE UGL I DAZI DAL DOBIJAMO TABELU

* PONOVITI ONA SLOVA O, 1, Z, H, W ...

ČESTA POJAVA U JAN

NE DOLAZI UGL U JANUARU ALI SVAKAKO TI TREBA !

Tabela 1. Instrukcije procesora

Operacija	Funkcija	Kod instrukcije
`mov Rz,Rx`	RZ ← [RX]	000000
`add Rz,Rx,Ry`	RZ ← [RX] + [RY]	000001
`sub Rz,Rx,Ry`	RZ ← [RX] - [RY]	000010
`and Rz,Rx,Ry`	RZ ← [RX] & [RY]	000011
`or Rz,Rx,Ry`	RZ ← [RX] | [RY]	000100
`not Rz,Rx`	RZ ← not [RX]	000101
`inc Rz,Rx`	RZ ← [RX] + 1	000110
`dec Rz,Rx`	RZ ← [RX] - 1	000111
`shl Rz,Rx`	RZ ← shl [RX]	001000
`shr Rz,Rx`	RZ ← shr [RX]	001001
`ashl Rz,Rx`	RZ ← ashl [RX]	001010
`ashr Rz,Rx`	RZ ← ashr [RX]	001011
`jmp ADDR`	PC ← ADDR	010000
`jmpz ADDR`	if Z=1 PC ← ADDR	010001
`jmps ADDR`	if S=1 PC ← ADDR	010010
`jmpc ADDR`	if C=1 PC ← ADDR	010011
`jmpnz ADDR`	if Z=0 PC ← ADDR	010101
`jmpns ADDR`	if S=0 PC ← ADDR	010110
`jmpnc ADDR`	if C=0 PC ← ADDR	010111
`ld Rz,Ry`	RZ ← [[RY]]	100000
`st Ry,Rx`	[RY] ← [RX]	110000

KONTROLNI ROM: → PRVO IDE FETCH INSTRUKCIJA !!!

dalji redosled nije bitan, bitno da ih nabrojimo

Sa slike (dato je)

ima 20 redova pamorabiti 5 bitu!

ADRESA	SADRŽAJ	INSTRUKCIJA
00000	5E3	Fetch !!!
00001	BE3	
00010	263	
00011	1B5	LDA
00100	2C3	
00101	5E7	
00110	2B3	SUB
00111	2E1	
01000	3CF	
01001	2BR	ADD
01010	2C1	
01011	3CD	
01100	1A3	INC
01101	2EC	
01110	3E3	
01111	3BE	SAL
10000	3CA	
10001	3C7	
10010	1A3	SAR
10011	2C3	
10100	3E3	

ADRESNI ROM: NEMA FETCH

PA JE PRVA SLEDEĆA SAD 1.

sadržaj te instr. je adresa na kojoj počinje sadržaj

ADRESA	SADRŽAJ	INSTRUKCIJA
0000 →	00011	LDA
0001	00110	SUB
0010	01001	ADD
0011	01100	INC
0100	01111	SAL
0101	10010	SAR
0110		
0111		
1000		
1001		
1010		
1011		
1100		
1101		
1110		
1111		

na ispitnom je dat kontrolni rom (tabela 1.)

U tabelama niže, data je sekvenca Fetch (prihvatne faze), kao i poslednje tri faze svake od instrukcija, obzirom da s mikroinstrukcije za svaku od njih upravo Fetch (prihvat).

Fetch (prihvat)	
Sekvenca	Kontr. reč
T1	5E3 Hex
T2	BE3 Hex
T3	263 Hex

SUB	
Sekvenca	Kontr. reč
T4	2B3 Hex
T5	2E1 Hex
T6	3CF Hex

INC	
Sekvenca	Kontr. reč
T4	1A3 Hex
T5	2EC Hex
T6	3E3 Hex

SAL	
Sekvenca	Kontr. reč
T4	3B[illegible] Hex
T5	3CA Hex
T6	3C7 Hex

LDA	
Sekvenca	Kontr. reč
T4	1B5 Hex
T5	2C3 Hex
T6	5E7 Hex

ADD	
Sekvenca	Kontr. reč
T4	2BR Hex
T5	2C1 Hex
T6	3CD Hex

SAR	
Sekvenca	Kontr. reč
T4	1A3 Hex
T5	2C3 Hex
T6	3E3 Hex

Vodeći računa o kodovima instrukcija iz Tabele 1, popuniti adresni ROM (Address ROM) i kontrolni ROM (Control ROM) sa Slike 1, tako da realizovana UJ ispravno radi.

Kontrolni ROM (Control ROM)

Adresa	Sadržaj	Instrukcija

Adresni ROM

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

Popuniti tabele tako da UJ radi ispravno, voditi računa o poziciji Fetch (prihvatne) mikrooperacije. Ukoliko postoje adrese u adresnom ROM-u čiji sadržaj nije bitan, popuniti ih sa XXXX ili ih ostavite prazne.

PONOVITI LARNOOVE MAPE !

* BINARNE KONVERZIJE * OBAVEZNO

ČIM PITA ŠTA TREBA DA SE URADI => DA SE PROŠIRI

^

KOD SABIRANJA

KOMPL 2:

1101 — 1° PROVERINO KOJI SU BROJEVI

+ 1010 — 1 + 4 − 8 = −3, → ILI -3

−9 — 2 − 8 = −6

TREBA

0111 = 7 ↯

1101

−

1

1100

0011

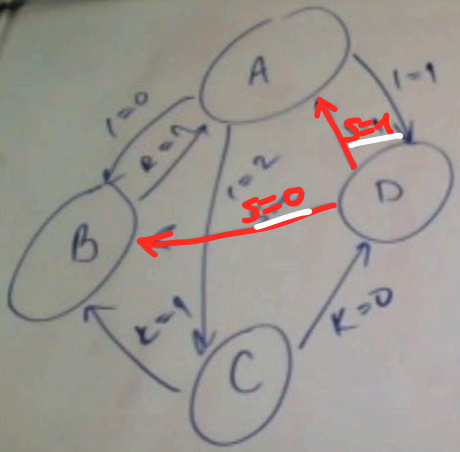
3

nmg da se saberu zbog prekoračenja

=> POTREBNO PROŠIRENJE:

10111 = −16 + 4 + 2 + 1 = −9 ✓

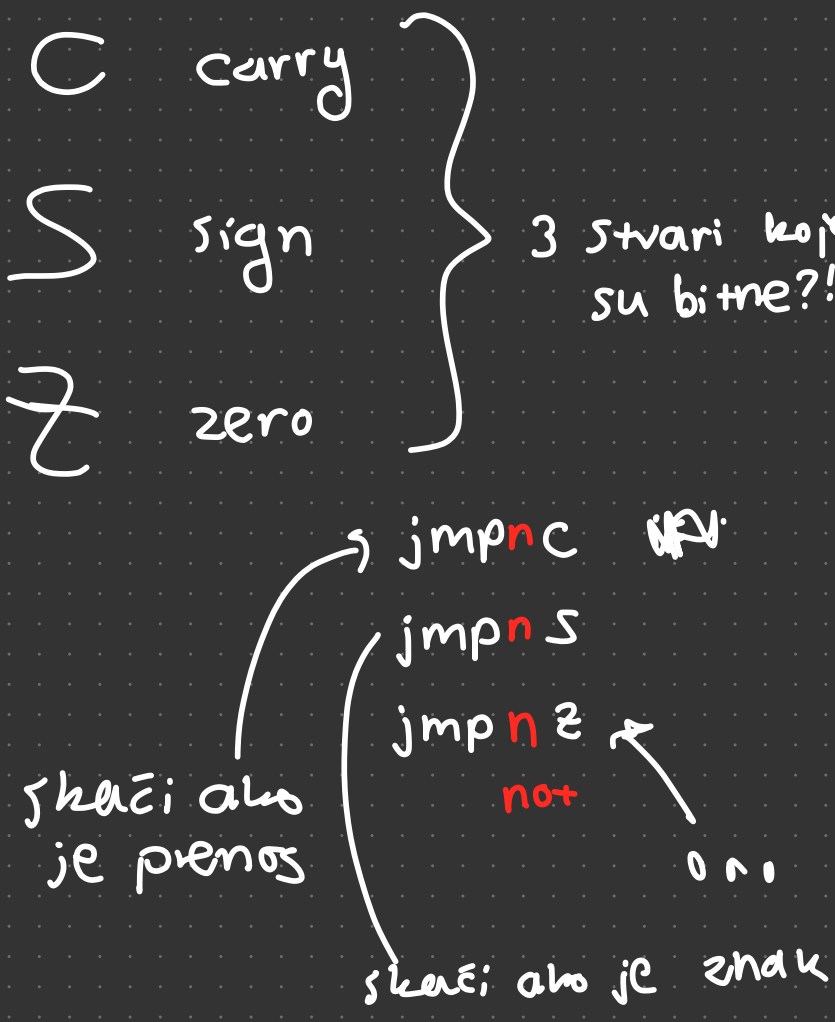
= jer je za to prekoračio




```
when X =>
    if(s=1) then
    SSTATE<= A;
    elsif(s=0) then
    SSTATE<=B
    else
        SSTATE<=IDLE;
    end if;
```

Stan X:

1) A
2) B
3) C
4) D


C carry
S sign
Z zero
3 stvari koje su bitne?!
jmpnc
jmpns
jmpnz
not
skači ako je prenos
skači ako je znak
0 ∧ 0