Eksamite kohta – näpunäited tudengile; õppejõududel lugemine keelatud!

- Eksam pole mingi loterii – keegi pole võitnud isegi raha, autost rääkimata.
- Ära õpi kõike järjest – teadus on piiritu, õpikuid on tuhandeid, aga pea on üks!
- Mis ei mahu pähe, talleta paberitükkidele, käelabale või joonlauale.
- Ära unusta õppejõudusid viisakalt tervitamast – see võib jäädagi eksami ainsaks meeldivaks hetkeks.
- Ära rutta piletit võtma. Kui sa midagi otsid, arvab õppejõud, et sa midagi ka tead.
- Ära alusta vastamist sõnadega „On üldteada, et...” Õppejõud võib mitte teada ning paluda sul täpsemaid selgitusi.
- Ära kunagi püüa meenutada seda, mida sa pole iial teadnud, muidu võid unustada sellegi, mida tead.
- Kasuta vastamisel ära kõik, mis peas on – niimoodi vabastad ajumahtu järgmise eksami jaoks.
- Ära kahtle oma vastustes! Las parem õppejõud kahtleb oma küsimustes.
- Ära kasuta tuntud aforismi „Ma tean seda, et ma midagi ei tea” – see on klassikute ja õppejõudude privileeg.
- Iga õppejõud soovib, et tudeng eksami läbiks, kuid mitte kõik ei tea, kuidas seda saavutada.
- Kui õppejõud küsib pärast vastamist „Mida te võiksida siia veel lisada?”, rääkige värske teemakohane anekdoot.
- Kui õppejõud ütleb „Head aega” vahetult pärast vastamist, tähendab see eksamil läbi kukkumist.
- Saades miinusmärgiga hinde, ära muretse – teaduses on miinus sageli etem kui pluss.

ISC0100 KÜBERELEKTROONIKA

Kevad 2025

Digitaalelektroonika analoogmaailmas (mäluta loogika)

Martin Jaanus NRG-308
martin.jaanus@ttu.ee 56 91 31 93

Õppetöö : http://isc.ttu.ee
Õppematerjalid : http://isc.ttu.ee/martin

Teemad

Digitaalelektroonika analoogmaailmas

- Lihtloogikafunktsioonid (eitus, või, välistav või, ning)
- Realisatsioon erinevatel tehnoloogiatel (DTL, TTL, KMOP....)
- Boole'i algebra
- Lihtloogikafunktsioonid –teisendamine.

Digitaalelektroonika

- Digitaalne (ladina keeles **digitis** – sõrm, inglise keeles digit- number) tähendab numbriline.
- Kasutusel (erinevalt pidevsüsteemist ehk analoogsüsteemist) kindlad , kokkulepitud signaali väärtused.
- Digisignaal saab muutuda astmete kaupa, lõplikud väärtused.
- Reeglina digitaalelektroonikas on digitaalne (kindlate väärtustega) ka aeg.

TAL
TECH

TUNNISTUS

Martin Jaanus

isikukoodiga 37706162218 on läbinud
Tallinna Tehnikaülikoolis täienduskoolituse

TAC0219 DIGITARKUS

15. oktoober 2018 – 28. veebruar 2019

ÕPIVÄLJUNDID

Koolituse läbinu:

- oskab luua lihtsat digitaalset sisu;
- oskab veebist vajalikku informatsiooni leida;
- on teadlik infoturbe põhitõdedest;
- on teadlik vajadusest pidevalt ajakohastada oma digitaalseid oskusi.

Maht: iseseisev töö 156 tundi (6 EAP)
Lektor: Jaanus Põial, Kaido Kikkas, Mario Metshein
Korraldaja: Avatud ülikool

Marika Tamm
Avatud ülikool
projektijuht

TTÜ

Tallinn, 15. märts 2019

TALLINNA TEHNIKAÜLIKOOL
REG KOOD 74000323

050008

Arvusüsteemidest

- On olemas positsioonilised ja mittepositsioonilised arvutussüsteemid.
- Mittepositsiooniline → Rooma süsteem.
- Positsioonilises süsteemis on tähtis numbri asukoht arvus.
- Suvaline arv X positsioonilises süsteemis alusega q üldjuhul võib esitada
- $X_q = X_{n-1} * q^{n-1} + X_{n-2} * q^{n-2} + \cdots + X_0 * q^0 + X_{-1} * q^{-1} + X_{-m} * q^{-m}$
- Kus X_i on järgutegur ($X_i = 0 \ldots q - 1$)
- q^i on kaalutegur ja q on süsteemi alus
- Igapäevaelus kasutame valdavalt süsteemi q→ 10
- Kuid…osade mõõtühikute puhul (näiteks aeg, ka 12, 24 ja 60 süsteemi)
- Digitaaltehnikas q→2 (binary 0,1) , 16 (hexadecimal 0..F) , 8 (octal 0..7))
- Teoreetiliselt on võimalik suvalise arvu (sh kompleksarvu või irratsionaalarvu nt e või π) baasil .

Aluse q valik

- Me tahame esitada mingit arvu, mis on omaette esitatud positsioonilises süsteemis, elektriliste signaalide abil. Seljuhul me vajame mingit elektrilist seadet, mis formeerib oma väljundil q erinevaid elektrilisi signaale. Kusjuures neid signaale peab oskama identifitseerida. Ja seda lihtsal viisil.
- Selliste seadmete kogus on võrdnejärkude hulgagaarvu terves ja murd osades.Selge see, et midasuurem on q, seda vähem läheb vaja seadmeid, aga seadmete keerukus kasvab tohutult.
- Aluse q valiku kriteerium – kulutuste vähendamine, häirekindluse säilitamine
- Optimaalseks osutub arv q=e=2.71.... Teostamine ülikeeruline ja mitteotstarbekas, seetõttu on valitud alus q=2

Arvusüsteemidest

- Näide 10 süsteemi naturaalarv $242 = 2 * 10^2 + 4 * 10^1 + 2 * 10^0$
- Teisendame 16 süsteemi (numbrid 0…F   10 - A, 11 - B, 12 - C ,13 - D,14 - E, 15 - F)
- Selleks jagame (leiame täisosa ja jäägi) arvusüsteemi alusega (16)
- 242:16= 15 , jääk 2 16^0
- 15:16=0, jääk 15 16^1
- Jagame, kuni jagatis on 0.
- Tulemus moodustub jääkidest ehk $15 * 16^1 + 2 * 16^0 = F2$ (hex)
- 15 asemel kirjutame F

Arvusüsteemidest

- Näide 10 süsteemi naturaalarv $242 = 2 * 10^2 + 4 * 10^1 + 2 * 10^0$
- Teisendame 2 süsteemi (numbrid 0-1)
- Selleks jagame (leiame täisosa ja läägi) arvusüsteemi alusega (2)
- 242:2= 121 , jääk 0 2^0
- 121:2= 60 , jääk 1 2^1
- 60:2= 30 , jääk 0 2^2
- 30:2= 15 , jääk 0 2^3
- 15:2= 7 , jääk 1 2^4
- 7:2= 3 , jääk 1 2^5
- 3:2= 1 , jääk 1 2^6
- 1:2= 0 , jääk 1 2^7
- Jagame, kuni jagatis on 0.Siit näha, et vaja 8 bitti .Tulemus moodustub jääkidest ehk
- $1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 11110010$ (bin)

Kahendsüsteem ja kuueteiskümnensüsteem

- On omavahel lihtsalt seotud ($16 = 2^4$) ja seepärast arvutites kasutusel . Tuleb vaid meelde jätta see tabel →

- Näide F2 (hex) kahendüsteemi
- F asemele 1111 ja 2 asemele 0010
- F2(hex)=11110010(bin)
- 2F 6A 33 (hex) → 00101111 01101010 00110011
- Saab ka vastupidi lihtsalt teisendada

- 10110011= (eraldame 4 kaupa bitid) 1011 0011=B3 (hex)
- Kümnendsüsteemi sedasi teisendada lihtsalt ei saa.

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

Kahendsüsteem

- Põhiline kasutus on kahendsüsteem , sest seda on lihtne kasutada (signaal kas on või ei ole)
- Kahe võimaliku oleku puhul on tegu binaarse signaaliga.
- Kõige lihtsam arvusüsteem. Bitt.
- 0 – vale, puudub, madal tase
- 1- tõene, olemas, kõrge tase

- Biti tähendus on kokkuleppeline , see võib tähendada mida iganes !

- Näiteks Arduino puhul a=digtalRead(1);
- Kui a väärtus on pärast seda käsku “true”, same vaid teada, et pinge oli klemmil “1” suurem kui 2 V . Milline see pinge tegelikult oli, selle kohta info puudub. Samas see pinge > 2V võib meie jaoks tähendada, et keegi astus tuppa ja andur selle registreeris.

Kahendsüsteem elektrilisel kujul

- Voolupõhine (tööstuselektroonika, -automaatika)

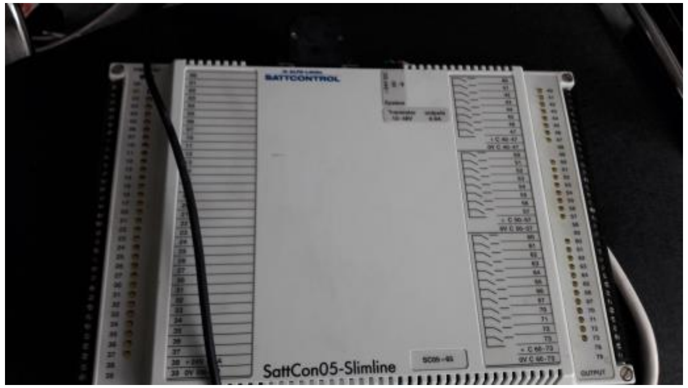
0 - 4 mA, 1 – 20 mA , kui vool puudub ,on ühendus katki.

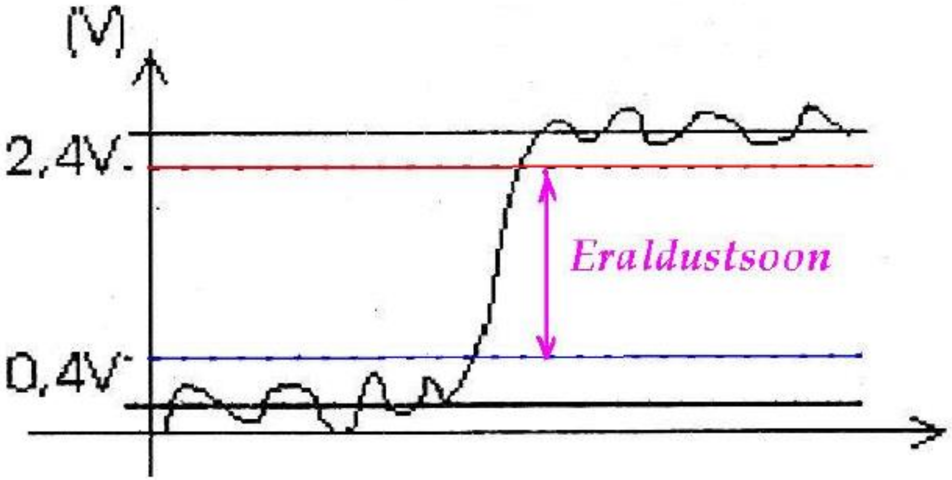
Valdavalt kasutatakse pingepõhist süsteemi

- 0 – 0...0.5 V   , 1 – 2.4.....(3.3 V , 5 V)

Tööstuselektroonikas ja häirerikkas keskkonnas kasutatakse ka teisi nivoosid 0 >>5 V, 1 < -5V...-24 V (RS232)

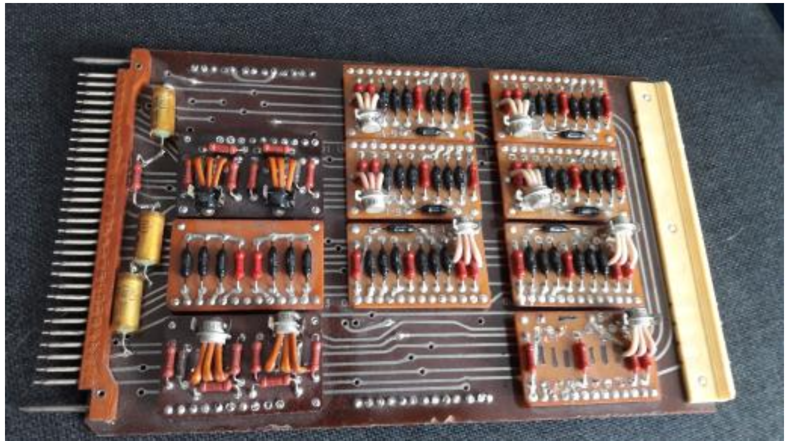
Aga olekuid võivad olla kodeeritud ka vahelduvpingesse (modulatsioon) : Amplituud,sagedus, faas. Tänapäevane sidetehnika.





Veidi ajaloost

- 1705 Binaarsüsteem (0,1) Gottfried Wilhelm Leibniz
- 1886 Georg Boole algebra (loogikatehted), releeloogika
- 1907 Audioni (elektronlambi) kasutamine NING-EI tehtes.
- 1924 tänapäevase loogikaelementide eelkäijad.
- 1941 Esimene elektrooniliselt programmeeritav automaat (Konrad Zuze, Z3), kasutas elektronlampe.
- 1953 Esimene täispooljuhtarvuti.
- 1958 Esimene loogikamikroskeem.



Tööstusautomaatika digijuhtmoodul (1973)
NSVL

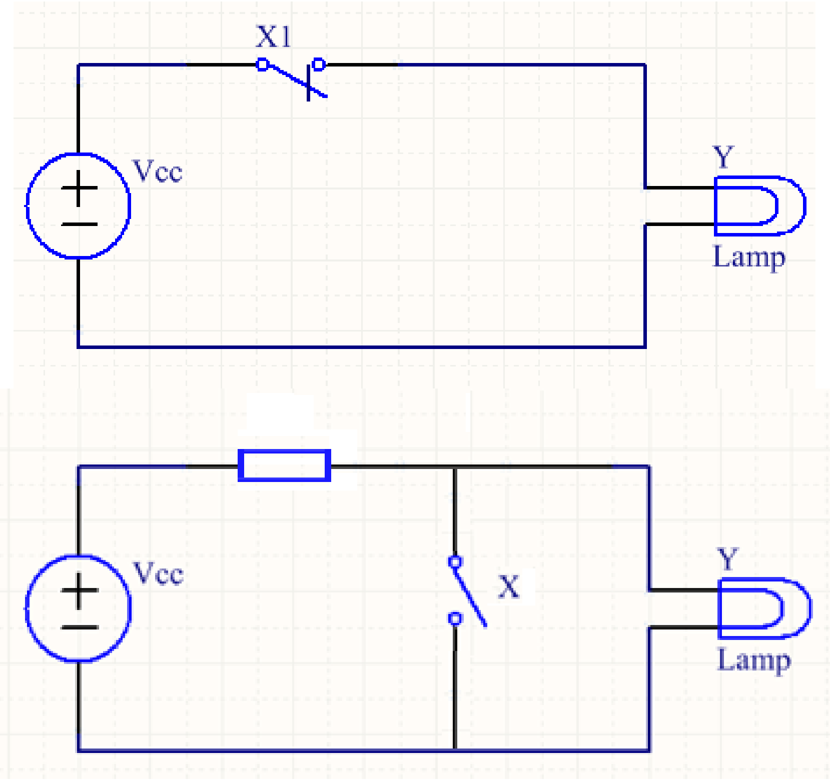
Tänapäev

- Lihtloogikaelementide masskasutus on vähenenud – neid asendavad mikroprotsessorid/kontrollerid. Igal juhul on need elemendid peidus selle sees.
- Diskreetelementidena kasutatakse – kui sedasi on lihtsam või töökiiruse pärast (programmi täitmine on aeglane)

Tööstusautomaatika digijuhtmoodul (1982) NSVL

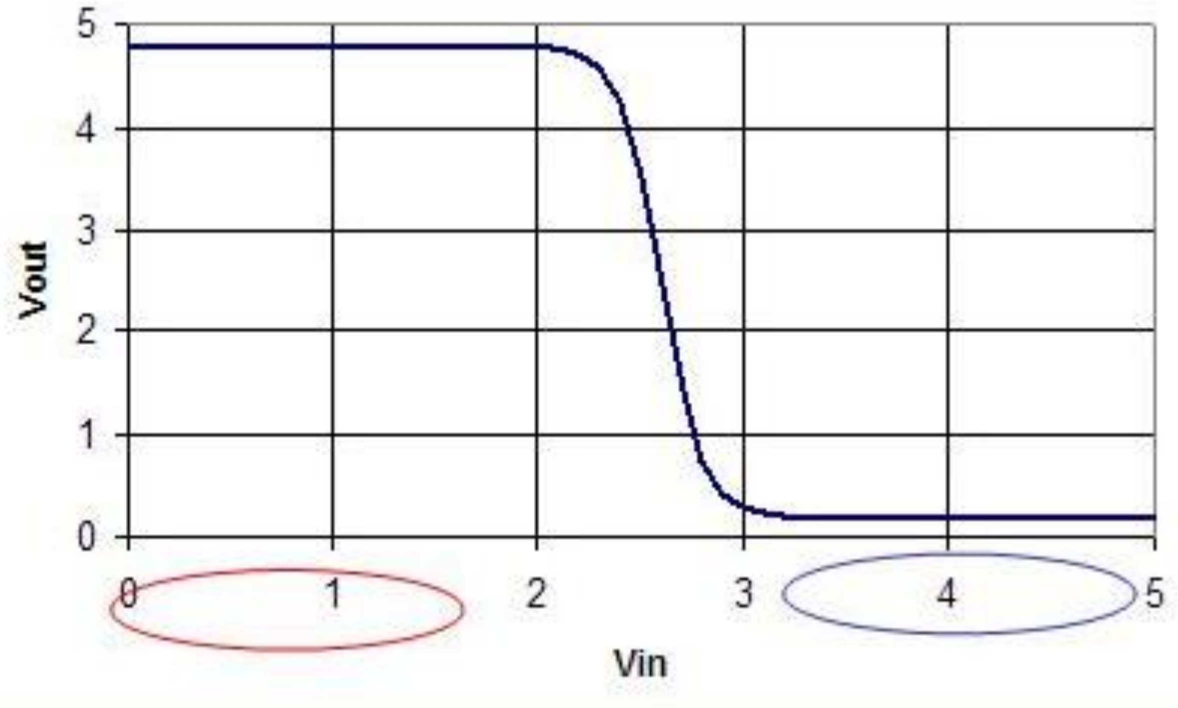
Loogiline tehe – eitus (inversioon)

- Loogikafunktsioon, ilma milleta ei ole digitaaltehnika võimalik !
- 0 → 1 ja 1 → 0 $Y=\overline{X}$



Kahendsüsteem elektrilisel kujul

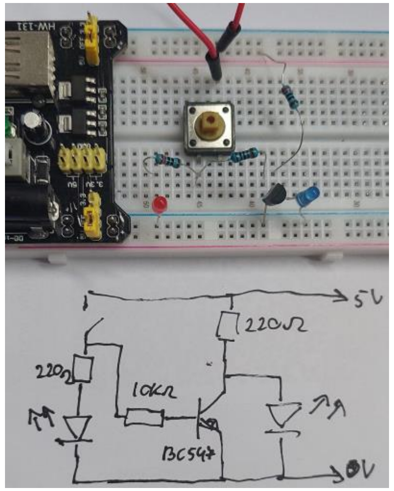
- Oluline on, et signaale töötlevad komponendid „kinnitaks“ olekut ehk viiks muutuja võimalikult kindlalt ja kiiresti vajalikku olekusse.

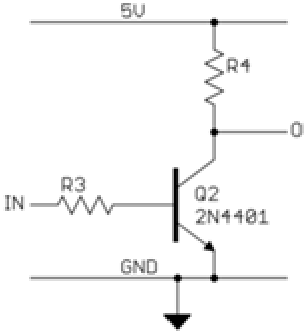


Inverteri olekuülekanne

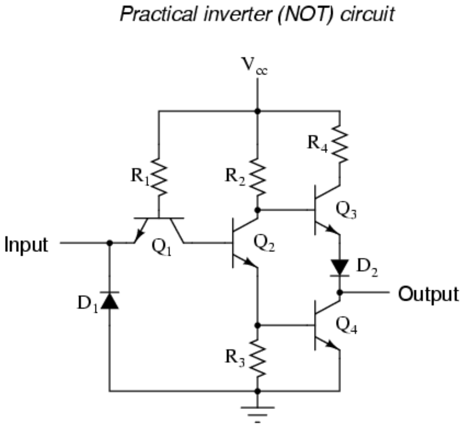
Loogiline tehe – eitus (inversioon)

- Lihtsam variant - saab teha ühe transistoriga.
- Transistor peab olema kas suletud või avatud (ei tohi olla aktiivrežiimis)

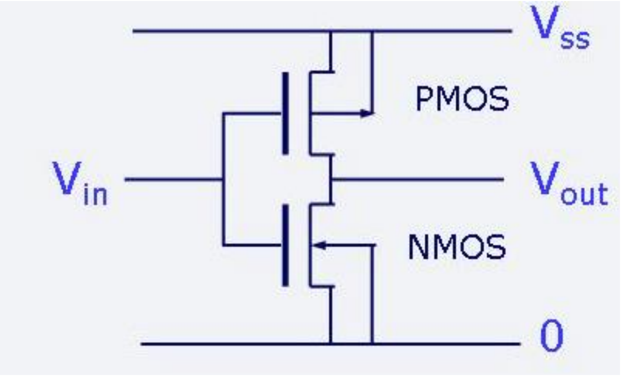




Diskreetelementidest



TTL



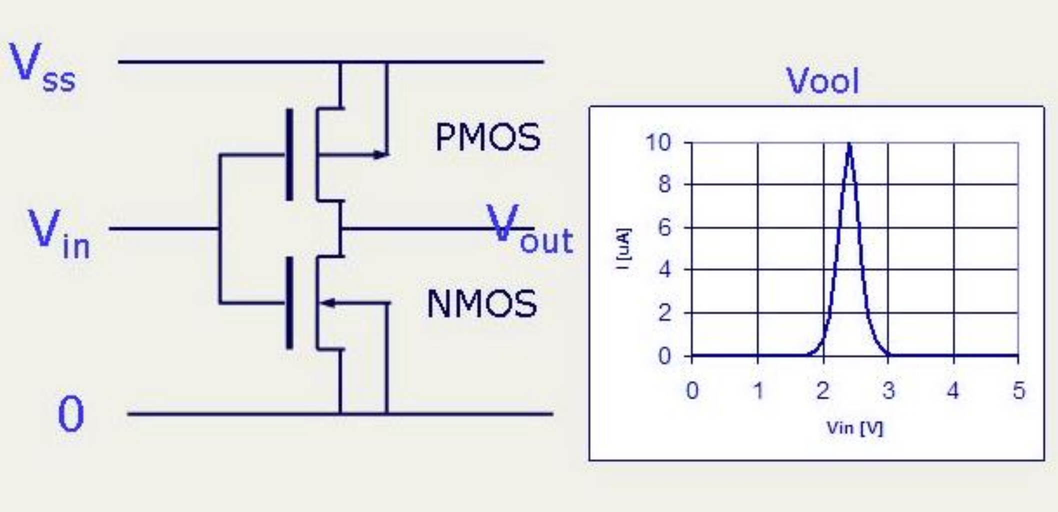
KMOP (CMOS)

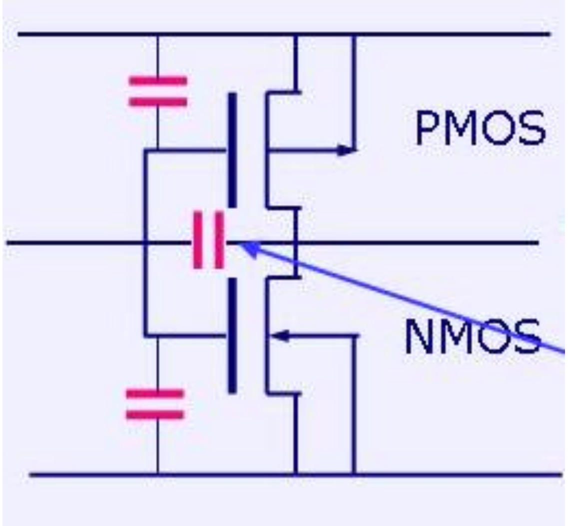
Mikroskeemisiseselt

Võib teha mitmel viisil aga populaarsem on CMOS tehnoloogia, mis ei tarbi oleku säilitamisel energiat.

Loogiline tehe – eitus (inversioon)

- Digitaaltehnika probleem – siire ühest olekust teise olgu võimalikult kiire !



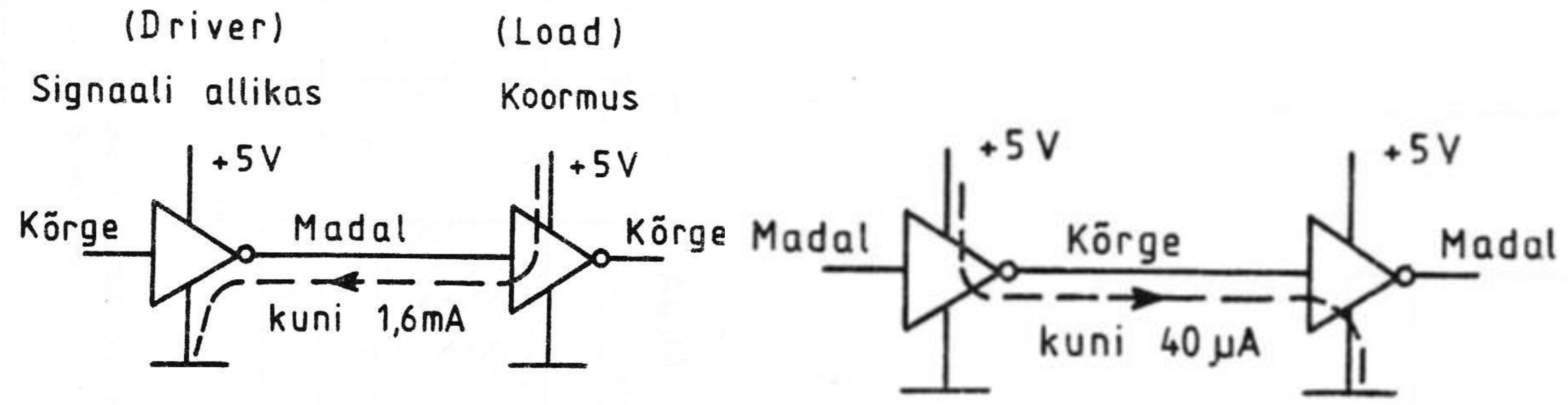


Toimub mahtuvuste ümberlaadimine.

- Lahendus – vähendame toitepinget (võimsus sõltub pinge ruudust) ning võimalusel töösagedust
- Jahutamine

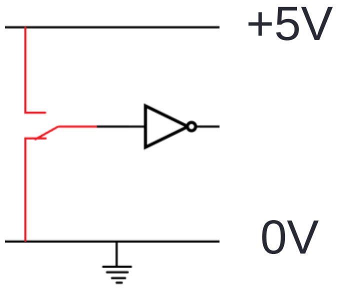
Loogikaelementide ühendamine

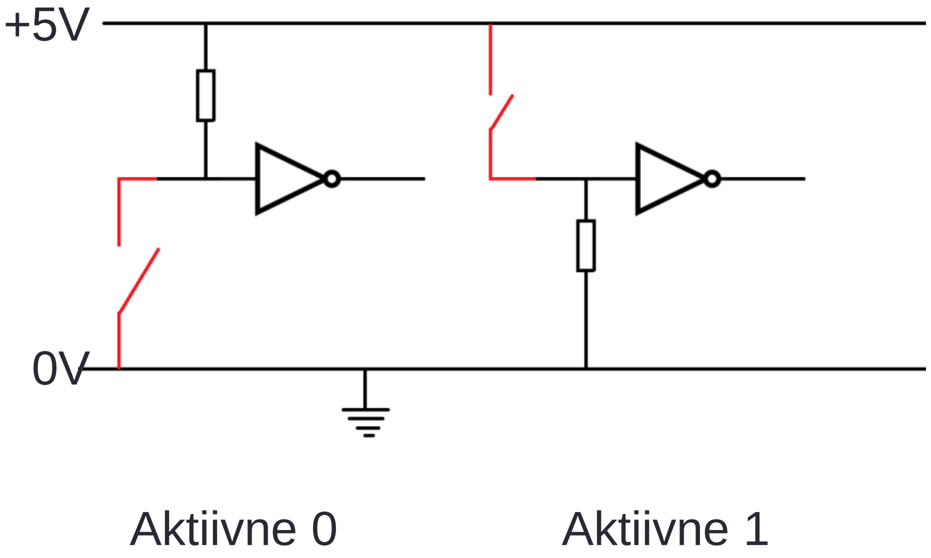
- Programmeerija jaoks on olemas vaid 0 ja 1
- Reaalses skeemis on pinged ning voolud – need on analoogsuurused ! Skeemide disainimisel tuleb sellega arvestada !
- Järgmine element tarbib voolu (CMOS ümberlülitusel, TTL pidevalt), elemendi väljund peab seda võimaldama.
- Üldjuhul saab väljundisse ühendada kuni 10 järgmise elemendi sisendit.



Loogikaelementide ühendamine

Ebamugav, →
kuigi vahest
kasutatakse →
(kui olek vaja
kinnitada)



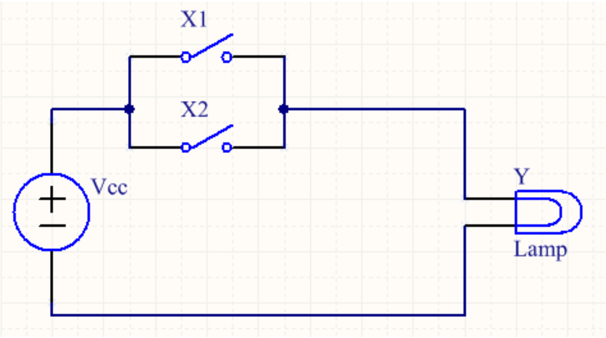


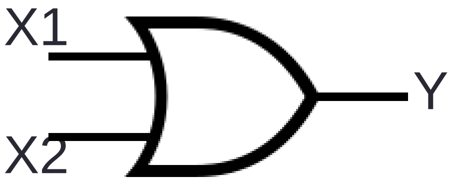
- Sisend ei tohi jääda avatuks (olek määramata, eriti CMOS tehnoloogias ,sh (mikro)kontrollerid) !
- Selle vältimiseks kasutatakse pinge toitesse ja maha tõmbavaid takisteid (pull-up ,pull down)
- Levinum variant on ülestõmbav takisti ja aktiivne olek “madal” (“maajuhe” on peeaegu alati käepärast)
- Võivad olla mikroskeemi sees (vt andmelehte !)
- Arduino : pinMode(klemm,INPUT_PULLUP);
- Ära kasuta mõlemat samaaegselt !
- Puudus – üleliigne energiatarve (takistite väärtused valida võimalikult suured)

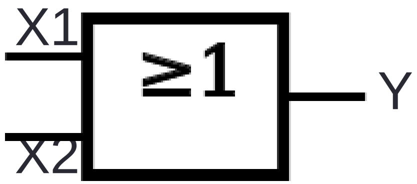
Takisteid ei ole vaja kui olek on eelmise elemendi poolt üheselt määratud !

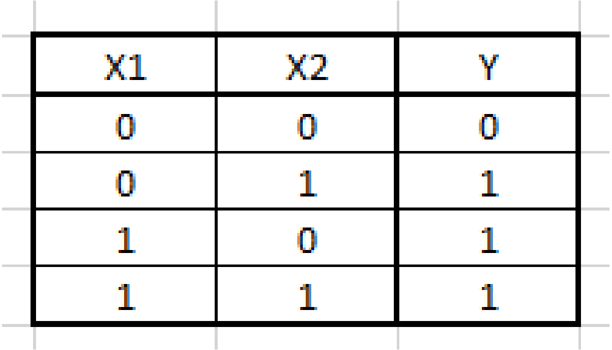
Loogiline tehe VÕI (OR)

- Vähemalt kahe sisendiga element.
- Väljund on 1 kui vähemalt üks sisend on 1.
- Y=X1+X2+....Xn







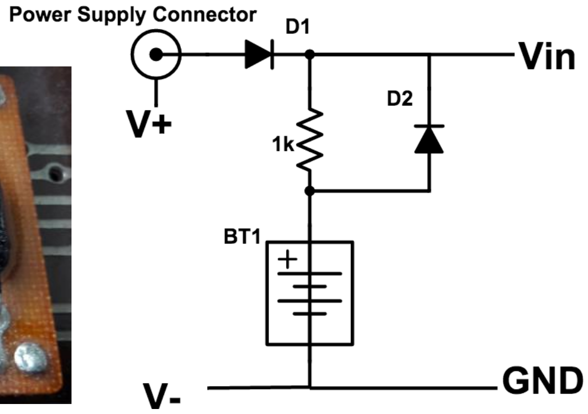


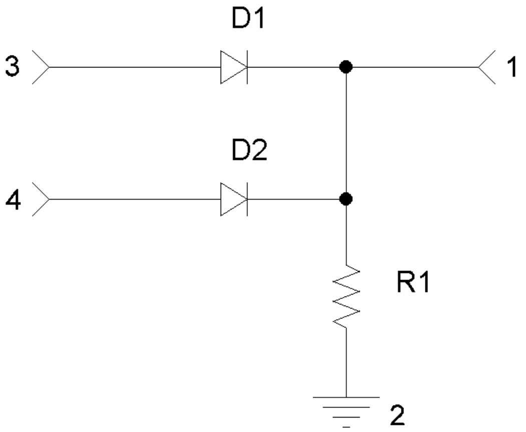
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Loogiline tehe VÕI (OR)

- Kõige lihtsam realisatsioon – kasutada dioode.
- Näide olmeelektroonikast - seadet toidab aku või elektrivõrk.
- Põhipuudus – dioodile jääb 0.7 V
- Kasutatakse tänapäeval diskreetelemente ja aeglastes kohtades . Mikroskeemisiseselt ei kasutata !

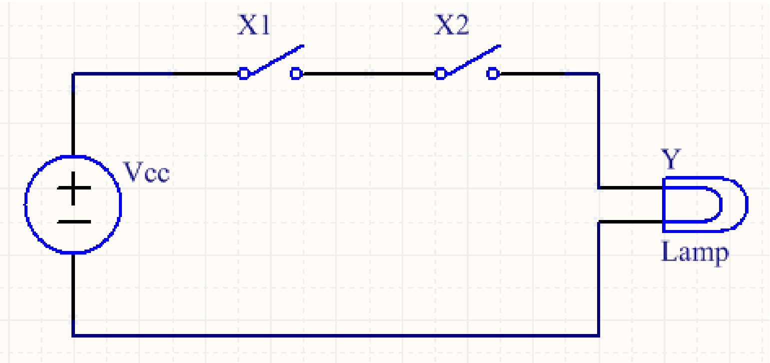
(NL -1973)

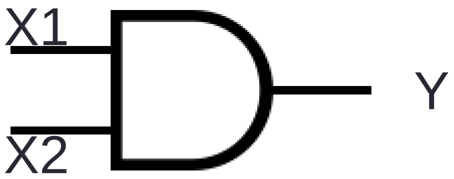


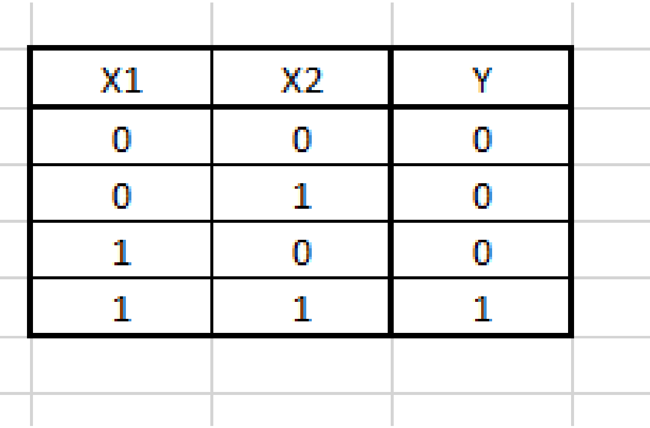


Loogiline tehe NING (AND)

- Vähemalt kahe sisendiga element.
- Väljund on 1 kui Kõik sisendid on samaaegselt 1.
- Y=X1*X2*....Xn



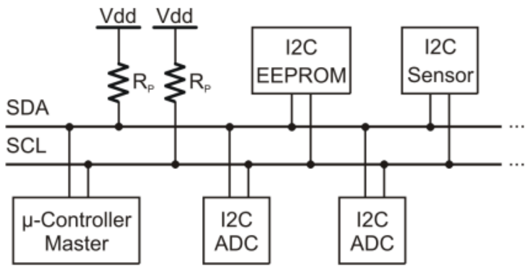


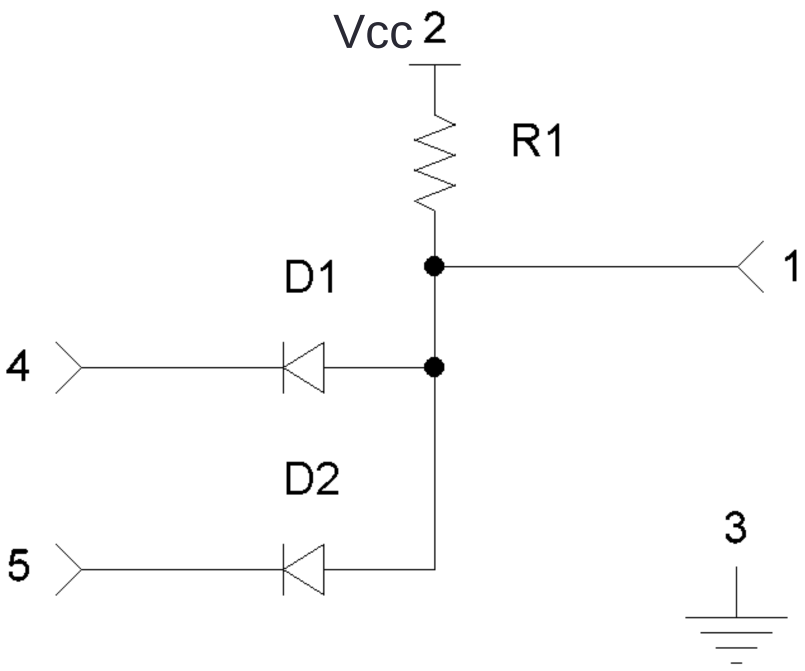


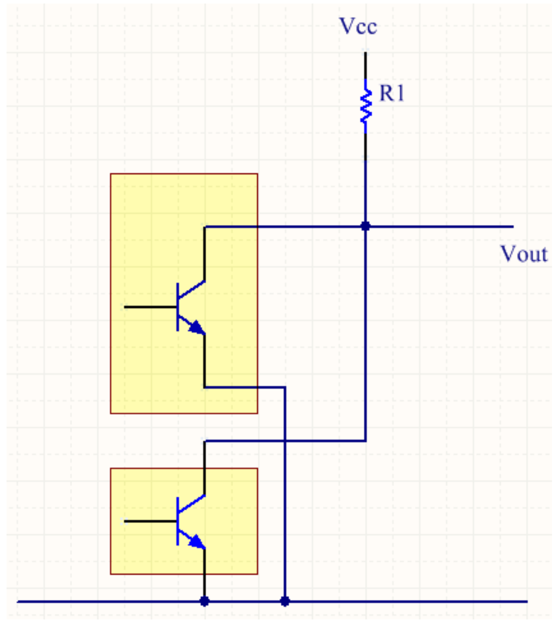
X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Loogiline tehe NING (AND)

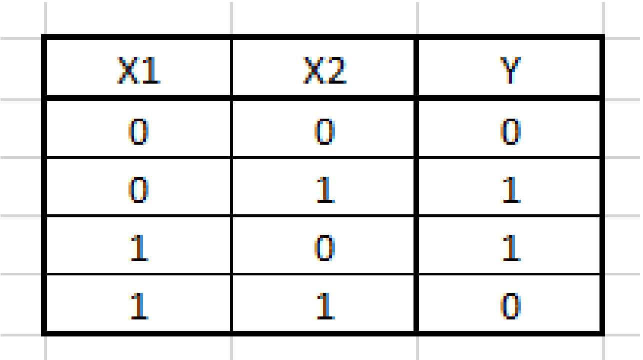
- Kõige lihtsam realisatsioon – kasutada dioode.
- See tehe toimub avatud kollektoriga elementide kokkuühendamisel.
- Andmesiinid (näit I2C)
- Mikroskeemi siseselt ei kasutata



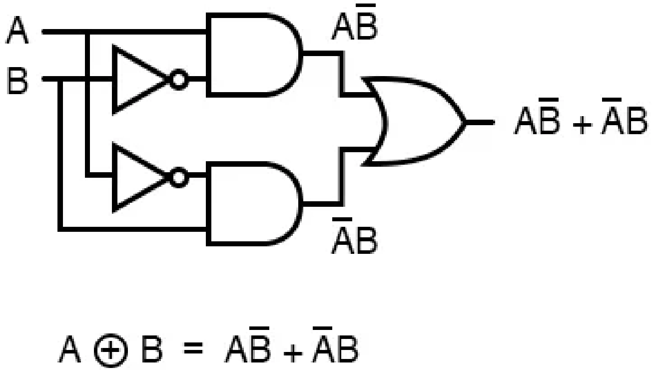




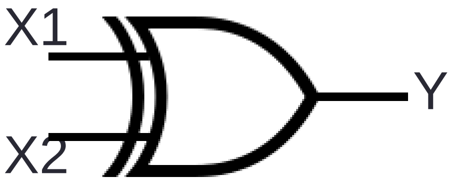
Välistav VÕI (XOR)

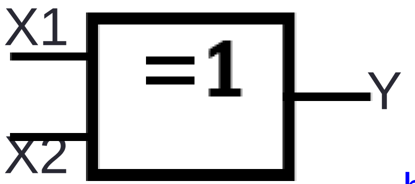


X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0



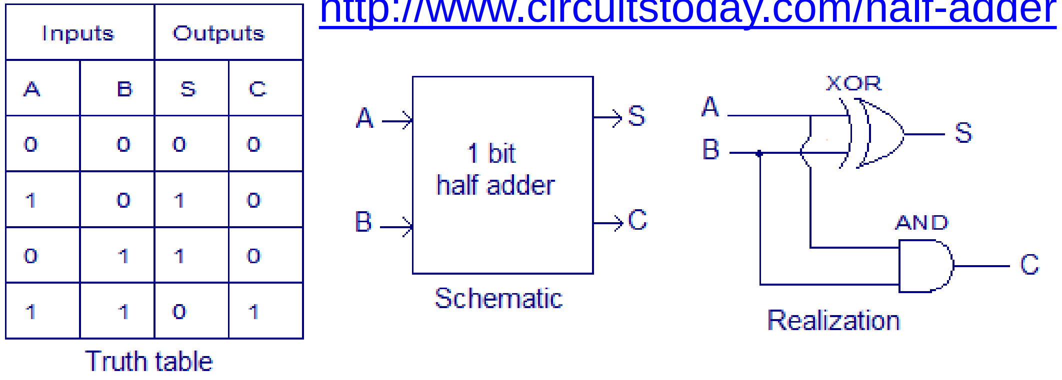
- Kahe sisendiga,võrdluselement
- Väljund on 1 kui sisendid on erinevad.
- $Y=X1\oplus X2$
- Ei ole elementaartehe .Saab teha VÕI-EI ja NING-EI elementidest.
- Põhiline kasutus protsessorites (summaatori koostisosa)





http://www.circuitstoday.com/half-adder

Poolsummaator →



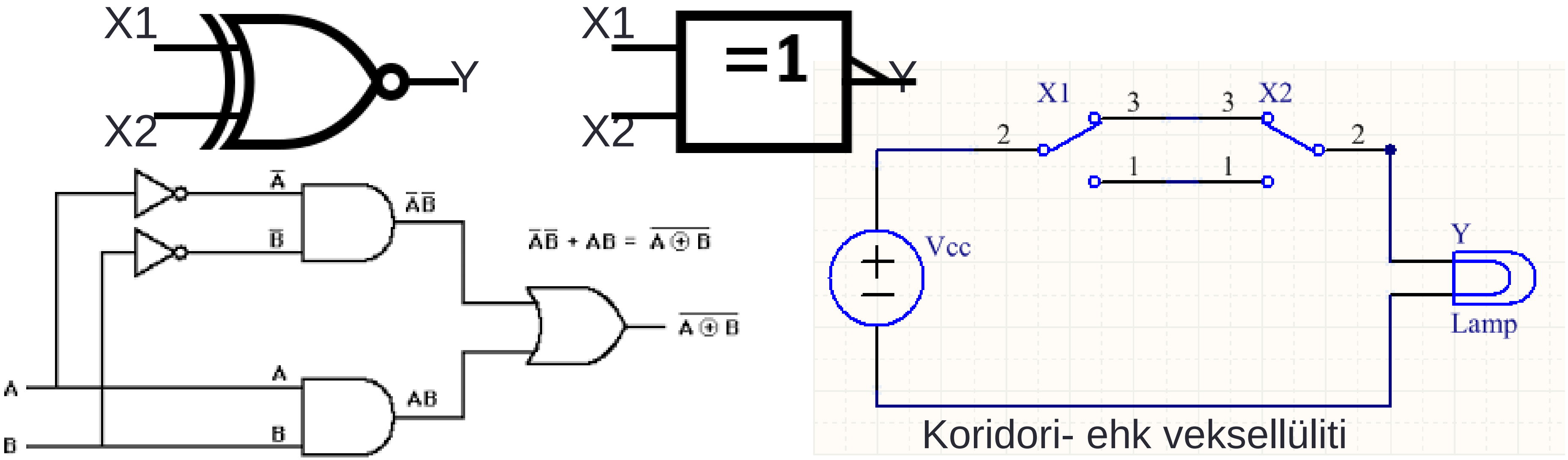
Inputs		Outputs	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Truth table

Välistav VÕI-EI (XNOR)

- Kahe sisendiga,võrdluselement
- Väljund on 1 kui sisendid on võrdsed.
- $Y=\overline{X1 \oplus X2}$
- Ei ole elementaartehe .Saab teha VÕI-EI ja NING-EI elementidest.
- Kasutatakse näiteks sünkroondetektoris,krüptograafias

X1	X2	Y
0	0	1
0	1	0
1	0	0
1	1	1



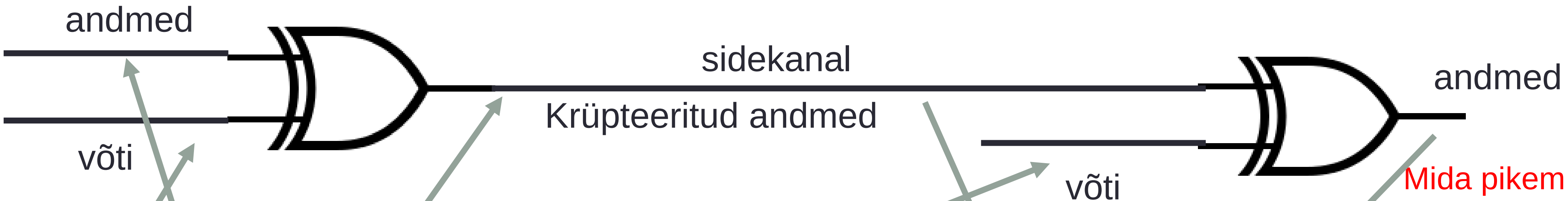
Koridori- ehk veksellüliti

XOR kasutamine

- Krüptograafia, tasuline TV , idee

Saadame läbi sidekanali 10 süsteemi arvud 178 ja 219
Kasutame ringlevat võtit 1100 (4 biti puhul 16 varianti)

Kui vastuvõtul kasutame mõnda teist võtit (näit. 1101), saame dekodeerimisel 163 ja 202
Kui dekodeerimisel võtit mitte kasutada, oleks tulemus 126 ja 23 (see liigub sidekanalis)



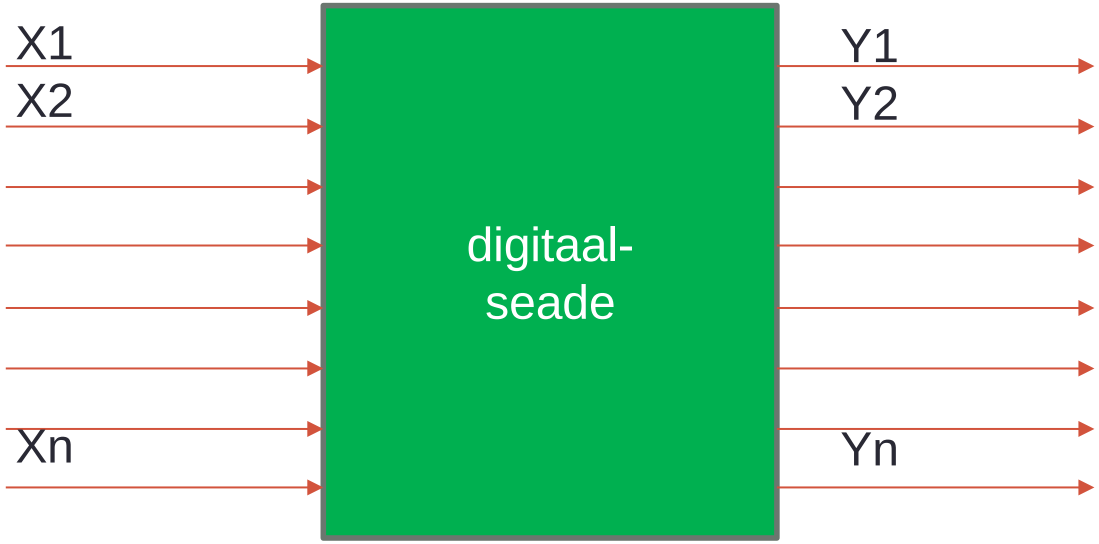
Mida pikem on võti, seda tülikam on seda proovimise teel ära arvata !

- XOR ja XNOR ei hävita infot vaid võivad teha sellele inversiooni
- Mõned bitid edastatakse invereeritult, millised, määrab ära võti .

võti:1100	(4 bitine, korduv)		ÕIGE võti:1100	vastuvõtja		VALE võti:1101	vastuvõtja	
	Sisend			Sisend			Sisend	
Info	10110010	11011011	sidekanalist::	01111110	00010111	sidekanalist::	01111110	00010111
Võti	11001100	11001100	Õige võti:	11001100	11001100	VALE võti	11011101	11011101
sidekanalisse:	01111110	00010111	dekodeeriud info:	10110010	11011011	dekodeeriud info:	10100011	11001010
			Sama ,mis enne krüpteerimist			Dekodeeritud info on vale !		

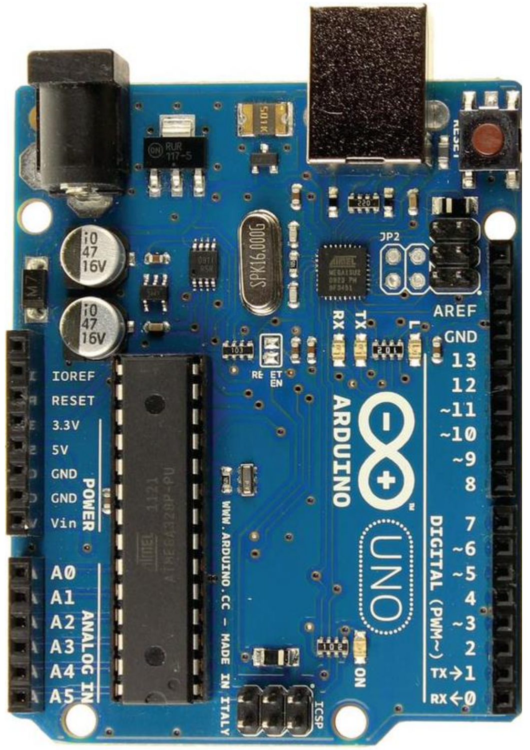
Loogikafunktsioon

- Digitaalse süsteemi käitumise kirjeldamine.
- (mäluta, mäluga, programmeeritav)
- Seob omavahel digitaalsed sisendid ja väljundid.

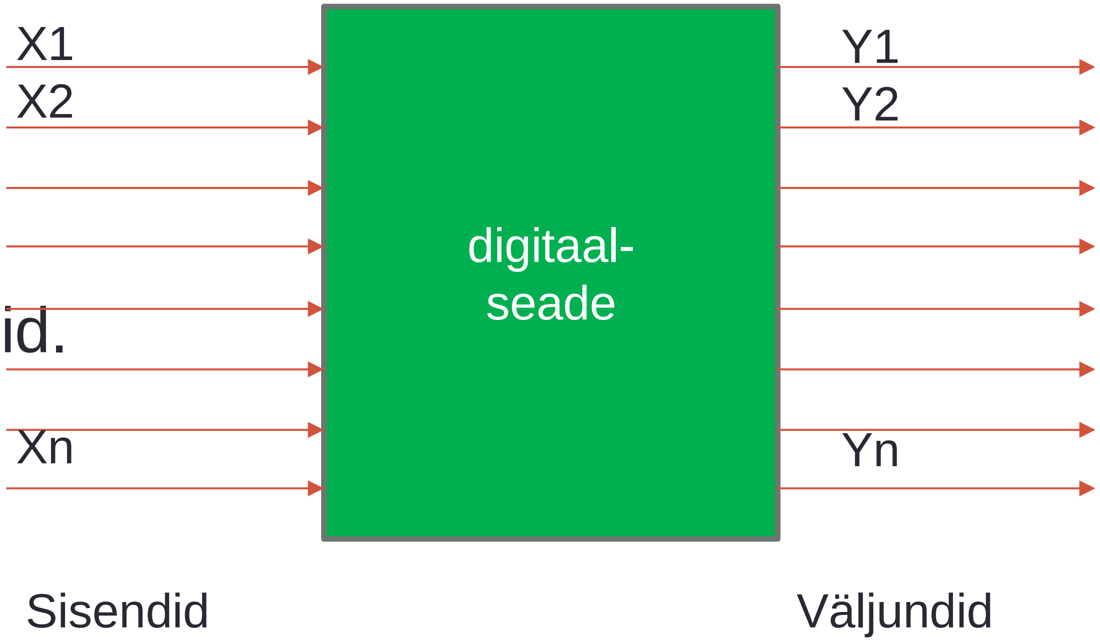


Sisendid

Väljundid



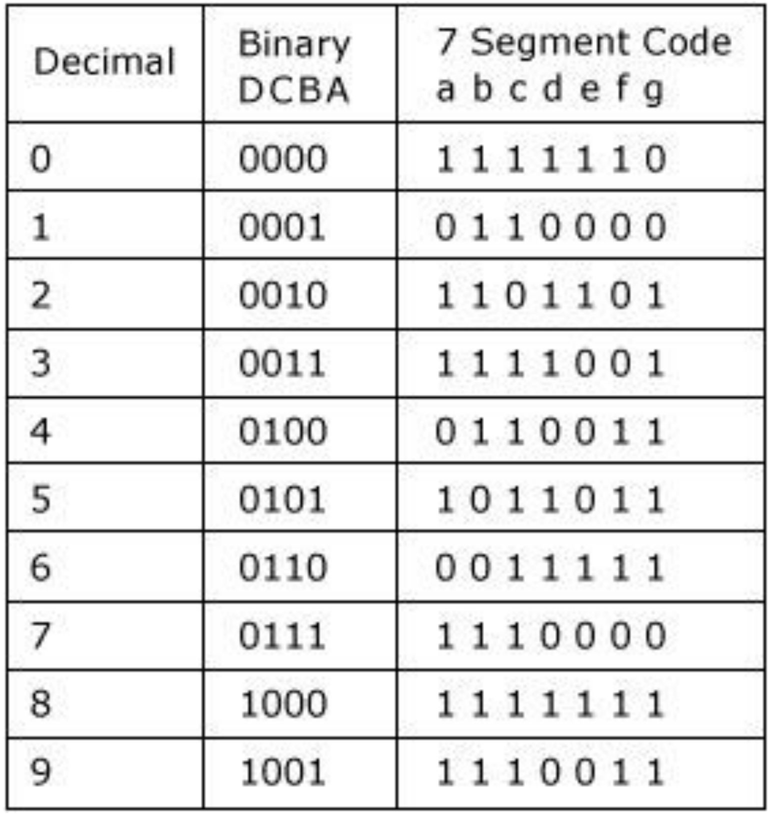
Mäluta loogikafunktsioon



- Digitaalse süsteemi käitumise kirjeldamine.
- Seob omavahel digitaalsed sisendid ja väljundid.
- Tuleb leida iga väljundi Y_i matemaatiline sõltuvus sisenditest $X_1 \ldots. X_n$.
- Kui sisendeid on n, siis iga väljund võib omada 2^n tähendust.
- Täielikult määratud loogikafunktsioonil ongi 2^n tähendust.
- Kui osa tähendusi puudub , on tegemist osaliselt määratud loogikafunktsiooniga.
- Sel juhul ei tohi osa sisendkoode esineda, sest väljundisse kujunevad fakultatiivsed (omavolilised) tähendused. Need sisendkoodid on keelatud .

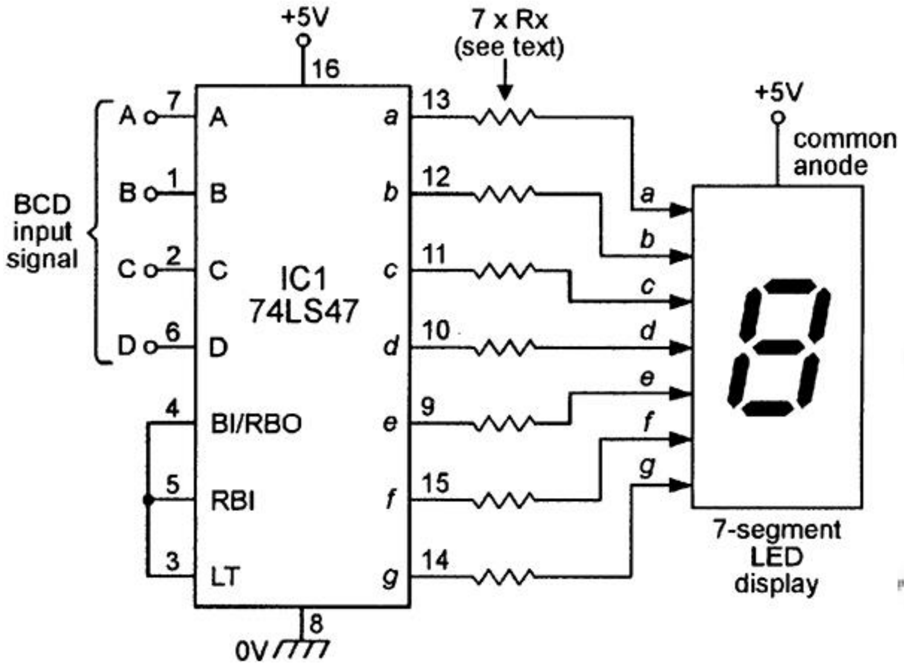
Kombinatsiooniloogika - Dekooder

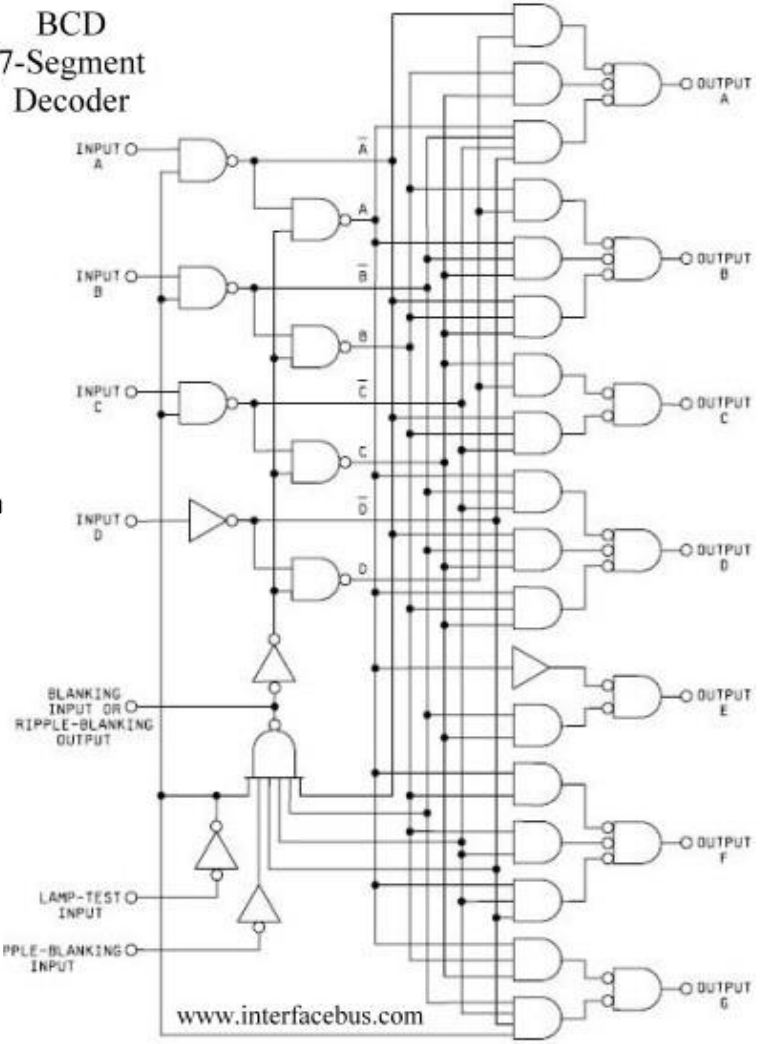
- Dekooder –lülitus, mis tunneb ära sisendisse saabuva kahendarvu ja annab signaali vastavasse väljundisse. Näiteks LED indikaatorite juhtlülitus.



Decimal	Binary DCBA	7 Segment Code a b c d e f g
0	0000	1 1 1 1 1 1 0
1	0001	0 1 1 0 0 0 0
2	0010	1 1 0 1 1 0 1
3	0011	1 1 1 1 0 0 1
4	0100	0 1 1 0 0 1 1
5	0101	1 0 1 1 0 1 1
6	0110	0 0 1 1 1 1 1
7	0111	1 1 1 0 0 0 0
8	1000	1 1 1 1 1 1 1
9	1001	1 1 1 0 0 1 1

Ülejäänud koodid on määramata !





http://www.interfacebus.com/ic-bcd-to-7-segment-decoder-schematic.html

Loogikafunktsiooni esitus tabelina

- Tõeväärtuste tabel – kõikvõimalikud sisendkombinatsioonid ja neile vastav väljundi väärtus
- Ülevaatlik, kuid ebamugav kasutada
- Sellest on lihtne kirjutada algebraline kuju.

X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Loogikafunktsioon algebralisel kujul (DNV)

- Disjunktiivne normaalne vorm (DNV) → elementaarsete loogikaliste korrutiste summa. NB! Elementaarsetes korrutistes argument või tema inversioon võib esineda ainult üks kord! DNV saab kätte tõeväärtuste tabelist

$$Y(X_2 X_1 X_0) = \overline{X}_2 X_1 X_0 + X_2 \overline{X}_1 X_0 + X_2 X_1 \overline{X}_0 + X_2 X_1 X_0$$

X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

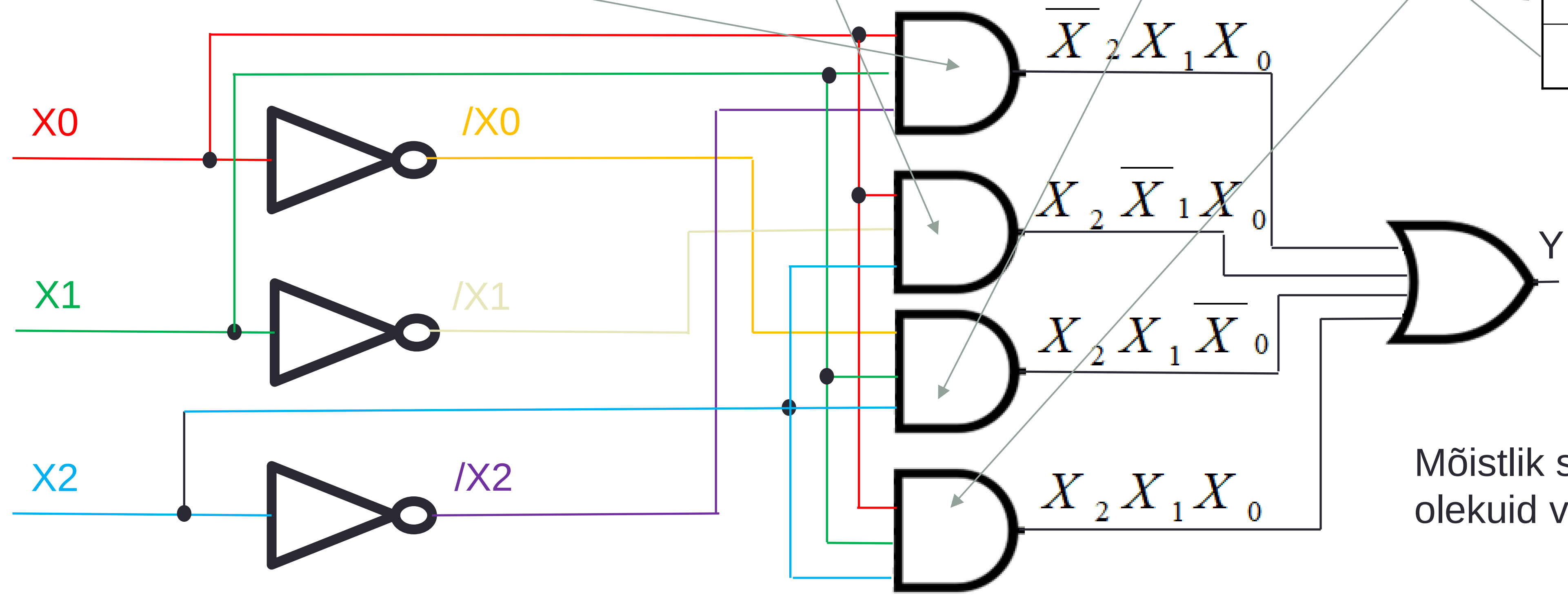
Koostamise reeglid:
Kirjutada välja tabelist X_{n-1}…..X_0 kõik kombinatsioonid, kus Y=1
Teha neist korrutised – ühe konstituendid 1→ X , 0 →/X
Liita kokku kõik ühe konstituendid.
See on TDNV – Täielik disjunktiivne normaalvorm

Loogikafunktsioonist – skeem DNV

- Algebralisel kujul olevast funktsioonist saab sünteesida skeemi.
- Praktiline kasutamine tülikas (enamasti ei toodeta selliseid loogikalülitusi)
- Toimib, kuid sageli ebamõistlik – saab lihtsustada .

X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

$$Y(X_2,X_1,X_0) = \overline{X}_2 X_1 X_0 + X_2 \overline{X}_1 X_0 + X_2 X_1 \overline{X}_0 + X_2 X_1 X_0$$



Mõistlik siis, kui väljundis 0 olekuid vähem kui 1 olekuid

Loogikafunktsioon algebralisel kujul (KNV)

- Konjunktiivne normaalne vorm (KNV) → elementaarsete loogikaliste summade korrutis. Koostame samuti tõesuse tabeli alusel. (negatiivne loogika)

$$Y(X_2 X_1 X_0) = (X_2 + X_1 + X_0)(X_2 + X_1 + \overline{X}_0)(X_2 + \overline{X}_1 + X_0)(\overline{X}_2 + X_1 + X_0)$$

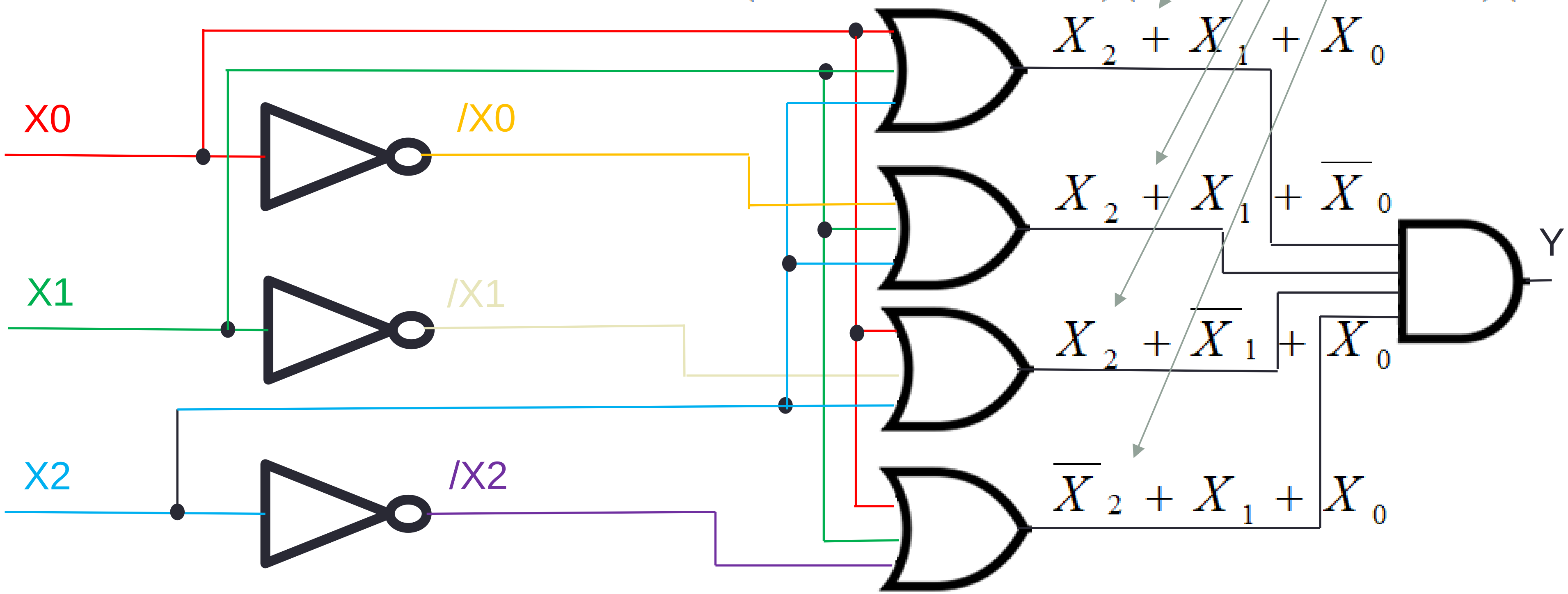
X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Koostamise reeglid:
Kirjutada välja tabelist X_{n-1}…..X_0 kõik kombinatsioonid, kus Y=0
Teha neist summad – nulli konstituendid 1→ /X , 0 →X
Korrutada kõik nulli konstituendid.
See on TKNV – Täielik konjuktktiivne normaalvorm

Täpselt sama funktsionaalsus

X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

$$Y(X_2 X_1 X_0) = (X_2 + X_1 + X_0)(X_2 + X_1 + \overline{X}_0)(X_2 + \overline{X}_1 + X_0)(\overline{X}_2 + X_1 + X_0)$$



Duaalsuse printsiip

Kui võrrelda tõesuse tabelit, mis vastavad tehtele NING ja VÕI, siis on kerge märgata, et kui tehe NING määravates tingimustes kõik sisendite ja funktsiooni enda tähendused vahetada nende inversioonide vastu, siis saame postulaadid, mis määravad VÕI tehe ja vastupidi (Postulaat – tõestuseta aktsepteeritav väide)

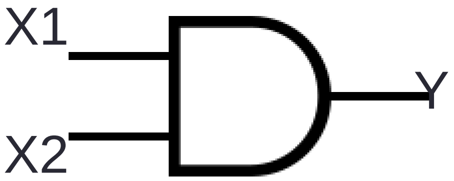
kui $X_1 * X_2 = Y$, siis $\overline{X_1} + \overline{X_2} = \overline{Y}$

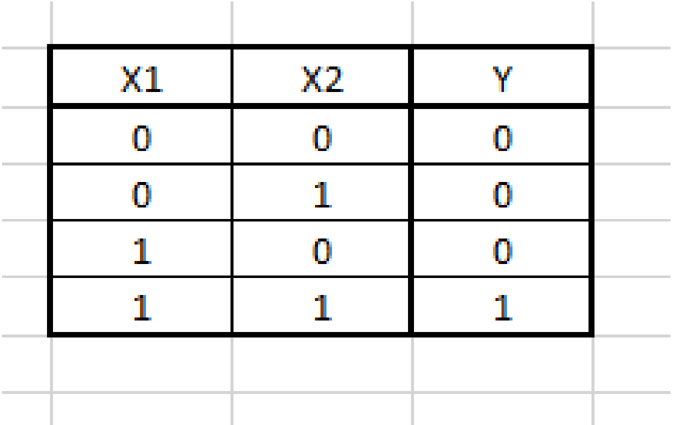
ja

kui $X_1 + X_2 = Y$, siis $\overline{X_1} * \overline{X_2} = \overline{Y}$

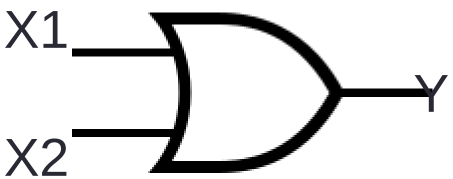
Funktsionaalselt täielik süsteem sisaldab eitust ja ühte kahest – NING või VÕI tehe.

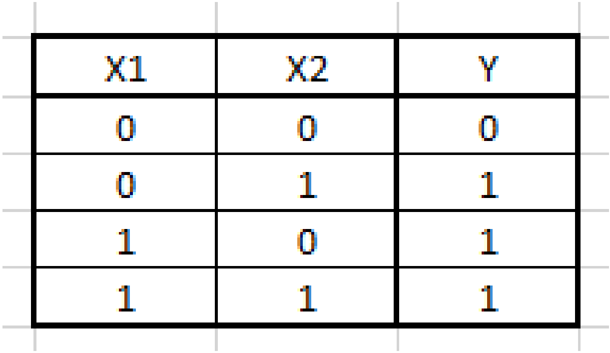
Seepärast on ongi populaarsed NING-EI ja VÕI-EI elemendid.





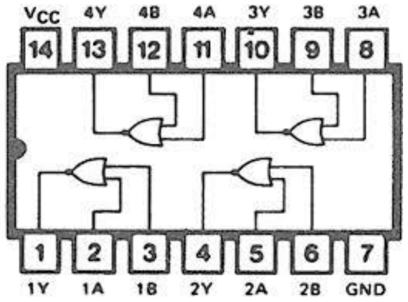
X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1





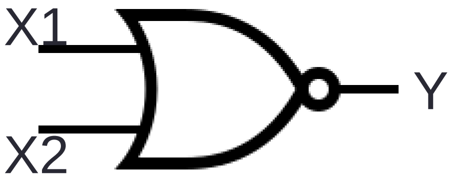
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

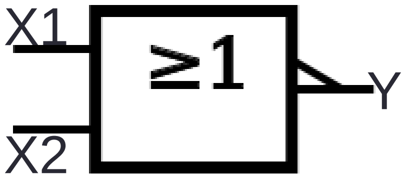
VÕI-EI (NOR)

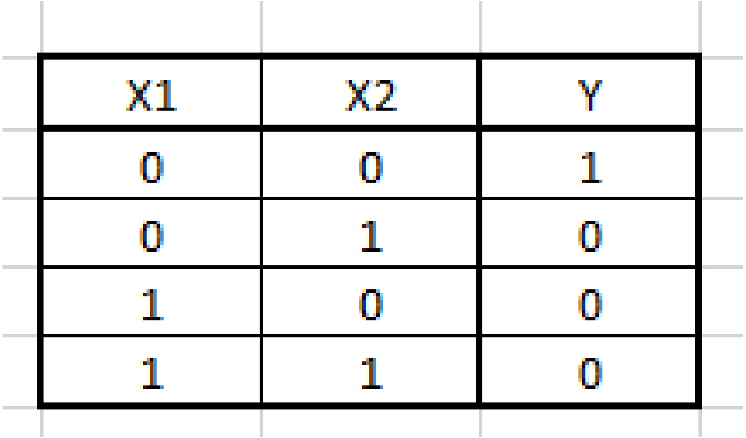


- Üks võimalikest loogika „põhiehituskivist“(vähemlevinud)
- VÕI ja EI elemendi kaskaadühendus.
- Väljund on 1 kui mõlemad sisendid on samaaegselt 0
- Y=X1+X2+....Xn Y=X1↓ X1 ↓ Xn

↓ - Peirce’i “nool” tähistab disreetses matemaatikas VÕI-EI tehet (Charles Sanders Peirce 1880)



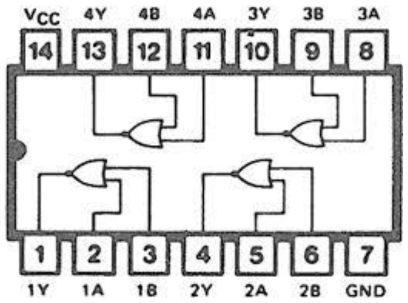


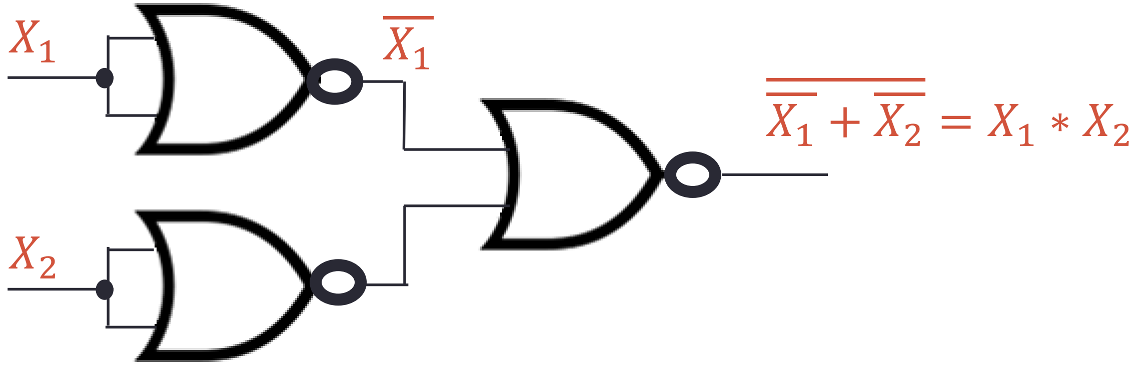


X1	X2	Y
0	0	1
0	1	0
1	0	0
1	1	0

- Ainult sellest elemendist piisab, et teha kõike !

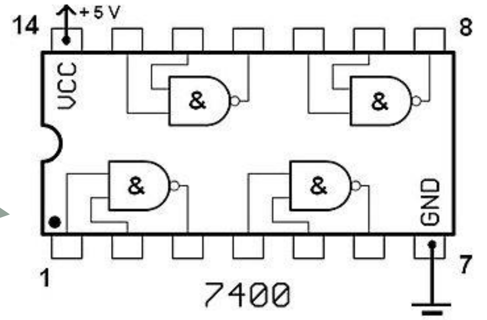
VÕI-EI elemendiga elementaartehted

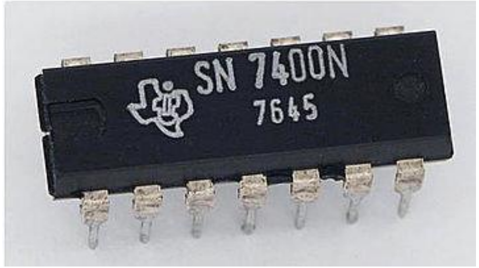




NING-EI (NAND)

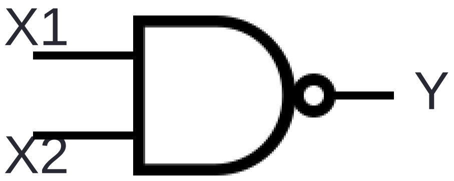
SN7400N alates 1963
Texas Instruments

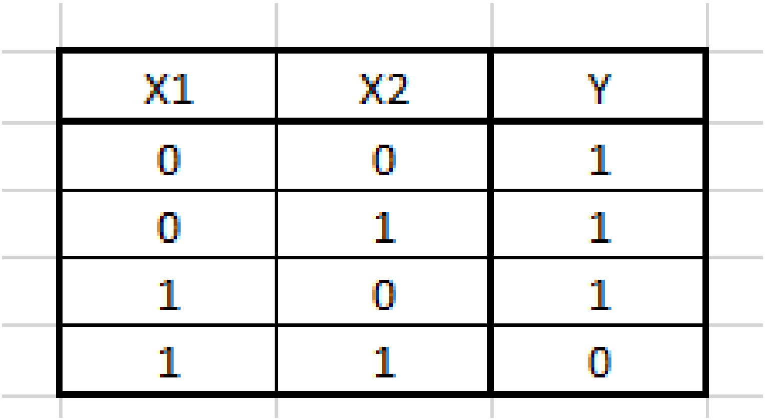




- Üks võimalikest loogika „põhiehituskivist“ (enamlevinud)
- NING ja EI elemendi kaskaadühendus.
- Väljund on 1 kui kasvõi üks sisend on 0
- Y=X1*X2*....Xn   Y= X1 | X2 | Xn

| - Shefferi “joon” – tähistab diskreetses matemaatikas NING-EI tehet

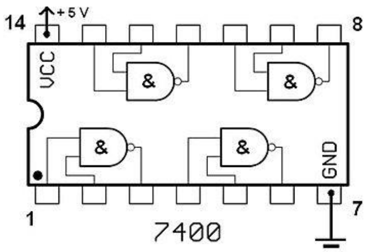


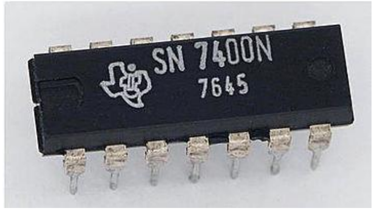


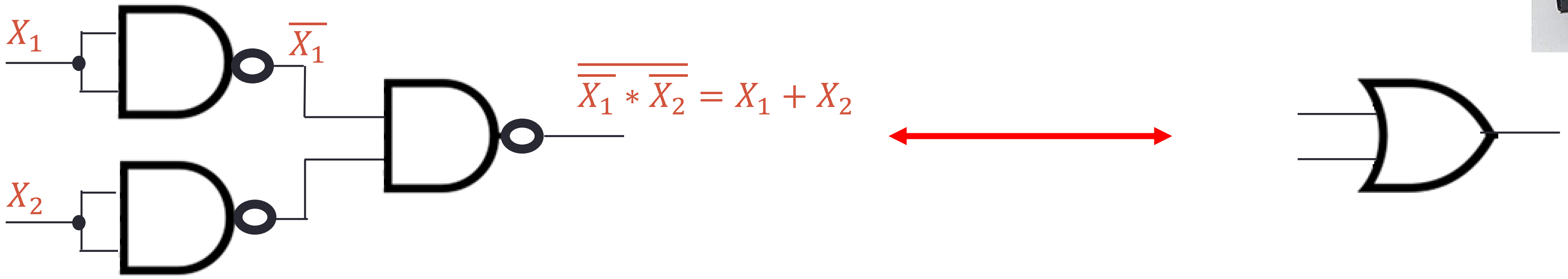
X1	X2	Y
0	0	1
0	1	1
1	0	1
1	1	0

- Ainult sellest elemendist piisab, et teha kõike !

NING-EI elemendiga elementaartehted







Boole’i algebra teoreemid

Kõik loogikaoperatsioonid alluvad duaalsuse printsiibile.

1) $X + 0 = X,$ $\qquad X \cdot 1 = X$

2) $X + 1 = 1,$ $\qquad X \cdot 0 = 0$

3) $X + X = X,$ $\qquad X \cdot X = X$

4) $\overline{X} + X = 1,$ $\qquad X \cdot \overline{X} = 0$

5) $\overline{\overline{X}} = X$ $\qquad$ ----------

6) $X_1 + X_0 = X_0 + X_1,$ $\qquad X_1 \cdot X_0 = X_0 X_1$

7) $(X_2 + X_1) + X_0 = X_2 + (X_1 + X_0),$

$(X_2 \cdot X_1) \cdot X_0 = X_2 \cdot (X_1 \cdot X_0)$

Boole'i algebra teoreemid

8) $\underline{\overline{X_1 + X_0} = \cdot \overline{X}_1 \cdot \overline{X}_0,}$ $\underline{\overline{X_1 \cdot X_0} = \overline{X}_1 + \overline{X}_0}$ → De Morgan

Võimaldab teisendust NING-EI ja VÕI-EI vahel.

→9) $X_1 \cdot X_0 + X_0 = X_0, \quad (X_1 + X_0) \cdot X_0 = X_0$

10) $X_2 \cdot X_1 + X_0 = (X_1 + X_0) \cdot (X_2 + X_0),$

$$(X_2 + X_1) \cdot X_0 = X_2 X_0 + X_1 X_0$$

11) $X_1 \cdot \overline{X}_0 + X_0 = X_1 + X_0,$

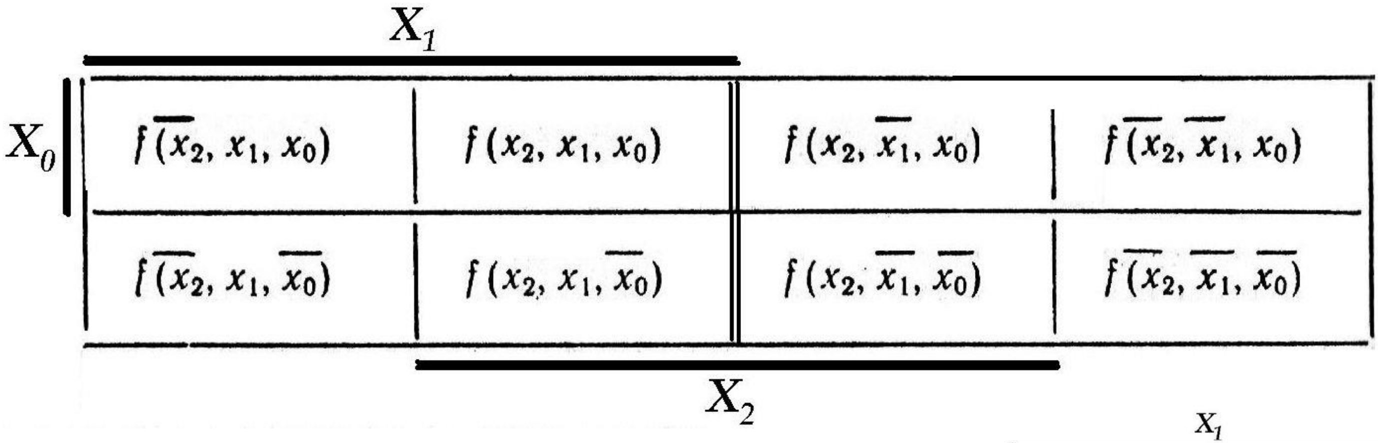
$$(X_1 + \overline{X}_0) \cdot X_0 = X_1 X_0$$

12) $X_1 \cdot X_0 + \overline{X}_1 \cdot X_0 = X_0$

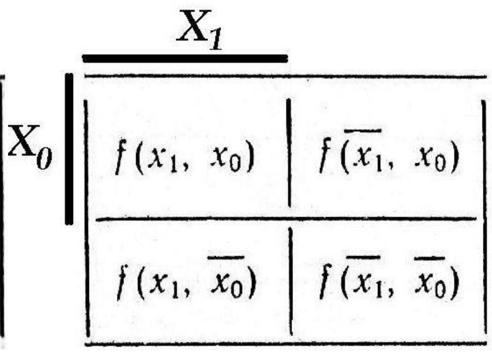
$$(X_1 + X_0) \cdot (\overline{X}_1 + X_0) = X_0$$

Täielikult määratud LF minimeerimine

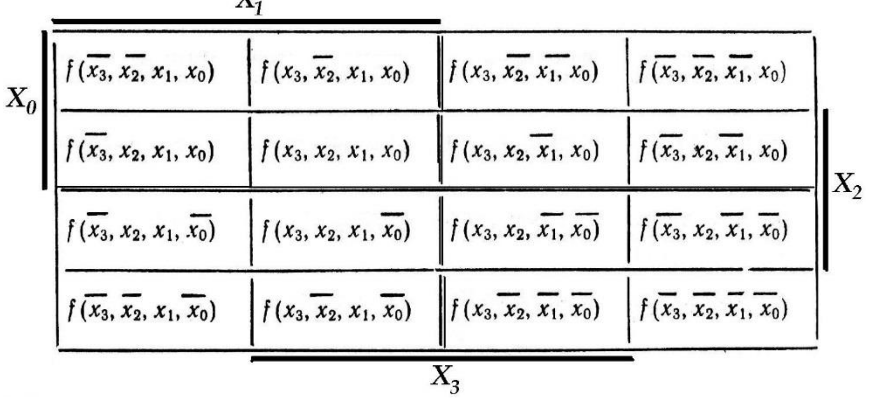
- Karnaugh (Veitch'i) kaardid: graafiline funktsiooni kirjeldus.
- Kaardid ja diagrammid → ruudulised tabelid; ruutude arv → **2^n**, kus n – muutujate hulk



$f(\overline{x_2}, x_1, x_0)$	$f(x_2, x_1, x_0)$	$f(x_2, \overline{x_1}, x_0)$	$f(\overline{x_2}, \overline{x_1}, x_0)$
$f(\overline{x_2}, x_1, \overline{x_0})$	$f(x_2, x_1, \overline{x_0})$	$f(x_2, \overline{x_1}, \overline{x_0})$	$f(\overline{x_2}, \overline{x_1}, \overline{x_0})$



$f(x_1, x_0)$	$f(\overline{x_1}, x_0)$
$f(x_1, \overline{x_0})$	$f(\overline{x_1}, \overline{x_0})$



$f(\overline{x_3}, \overline{x_2}, x_1, x_0)$	$f(x_3, \overline{x_2}, x_1, x_0)$	$f(x_3, \overline{x_2}, \overline{x_1}, x_0)$	$f(\overline{x_3}, \overline{x_2}, \overline{x_1}, x_0)$
$f(\overline{x_3}, x_2, x_1, x_0)$	$f(x_3, x_2, x_1, x_0)$	$f(x_3, x_2, \overline{x_1}, x_0)$	$f(\overline{x_3}, x_2, \overline{x_1}, x_0)$
$f(\overline{x_3}, x_2, x_1, \overline{x_0})$	$f(x_3, x_2, x_1, \overline{x_0})$	$f(x_3, x_2, \overline{x_1}, \overline{x_0})$	$f(\overline{x_3}, x_2, \overline{x_1}, \overline{x_0})$
$f(\overline{x_3}, \overline{x_2}, x_1, \overline{x_0})$	$f(x_3, \overline{x_2}, x_1, \overline{x_0})$	$f(x_3, \overline{x_2}, \overline{x_1}, \overline{x_0})$	$f(\overline{x_3}, \overline{x_2}, \overline{x_1}, \overline{x_0})$

Realistlik kasutada kuni nelja sisendi puhul

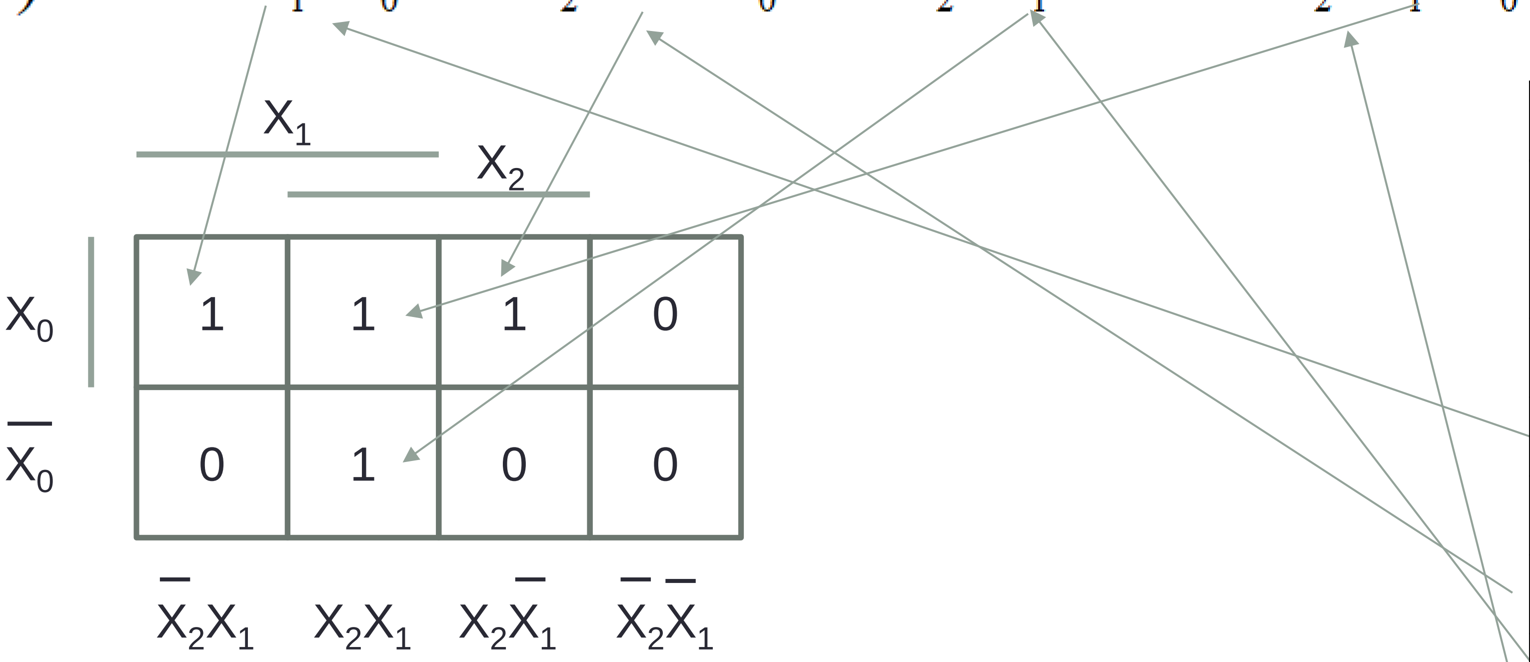
Täielikult määratud LF minimeerimine

- Veitch`i (Karnauh) kaardile on peale kantud n-muutuvatega LF. Tuleb välja valida, eraldada, täisnurgalised piirkonnad (kontuurid, katted), mis ühendavad kõik LF tähendused (loogika 1 või 0 järgi).
- Iga piirkond, kontuur, peab sisaldama $\mathbf{2^k}$ ruutu, kus **k** – täisarv.
- Eraldatud piirkonnad võivad ristuda. Teisiti: mõned ruudud võivad kuuluda erinevatele piirkondadele (kontuuridele).
- Saadud kontuuridest valida minimaalne arv maksimaalselt suuri kontuure, mis sisaldavad kõik LF tähendused.
- Loogikaliselt summeerida implekantid, mis vastavad valitud kontuuridele. Saadud summa ongi minimaalne disjunktiivne normaalne vorm (**MDNV**) juhul, kui kirjeldatud protseduur oli tehtud 1 – järgi.
- Kuna minimeeritav LF oli täielikult määratud, siis juhul, kui protseduur oli läbiviidud 0 – järgi, tulemuseks on minimaalne konjuktiivne normaalne vorm (MKNV).

Täielikult määratud LF minimeerimine

- Täidame Karnaugh ‘ , Veich’ I kaardi
- Saab ka otse olekutabelist

$$Y(X_2X_1X_0) = \overline{X}_2X_1X_0 + X_2\overline{X}_1X_0 + X_2X_1\overline{X}_0 + X_2X_1X_0$$



	$\overline{X_2}X_1$	X_2X_1	$X_2\overline{X_1}$	$\overline{X_2}\,\overline{X_1}$
X_0	1	1	1	0
$\overline{X_0}$	0	1	0	0

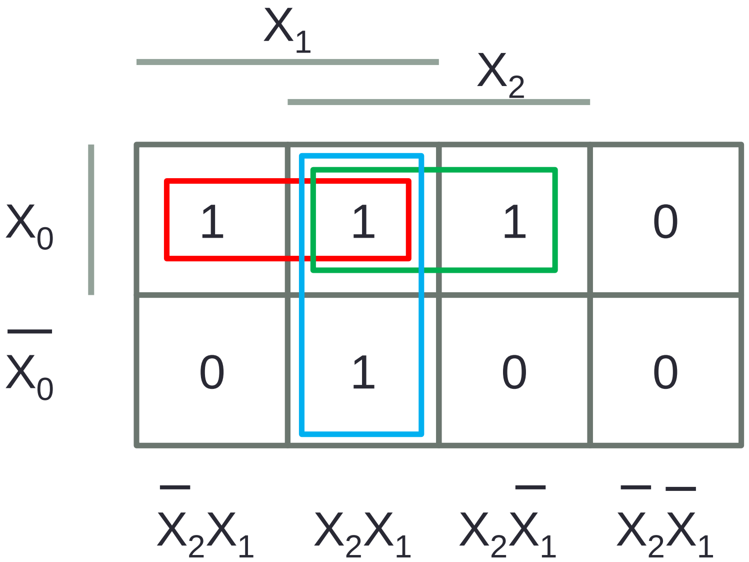
X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Täielikult määratud LF minimeerimine

- Leiame 2^k ühisosad

X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

$$Y(X_2 X_1 X_0) = \overline{X}_2 X_1 X_0 + X_2 \overline{X}_1 X_0 + X_2 X_1 \overline{X}_0 + X_2 X_1 X_0$$



	$\overline{X}_2 X_1$	$X_2 X_1$	$X_2 \overline{X}_1$	$\overline{X}_2 \overline{X}_1$
X_0	1	1	1	0
$\overline{X}_0$	0	1	0	0

$$Y = X_1 X_0 + X_2 X_0 + X_2 X_1$$

Siin on X_2 eitusega ja ilma

Võivad tekkida ka 4 kaupa , siin näites mitte.

Loogikaseadmete süntees etteantud baasi alusel

- Funktsionaalselt täieliku süsteemi NING, VÕI, EI tavaliselt ei kasutata.
- Praktikas kasutakse NING-EI, VÕI-EI või isegi ainult ühte nendest.
- NING-EI, VÕI-EI abil saab esitada ükskõik, millist LF !

Sünteesimisel on kaks võtet

- kahekordne inverteerimine (terve LF või osaliselt)
- De Morgan`i teoreemide kasutamine

Talutavad on kuni 4 muutujaga funktsioonid

Mõistlik lasta arvutil sünteesida (kuigi arvuti ei pruugi anda mõnikord mõistlikku lahendust)
Keerukus kasvab kiiresti muutujate arvu suurenedes.

Loogikaseadmete süntees etteantud baasi alusel

- Näide : Kasutada on vaid NING-EI elemendid

NB ! Negatiivse loogika VÕI tehe.
Väljund on 0 eitus (1) kui esimene sisend
VÕI teine sisend VÕI kolmas sisend on 0

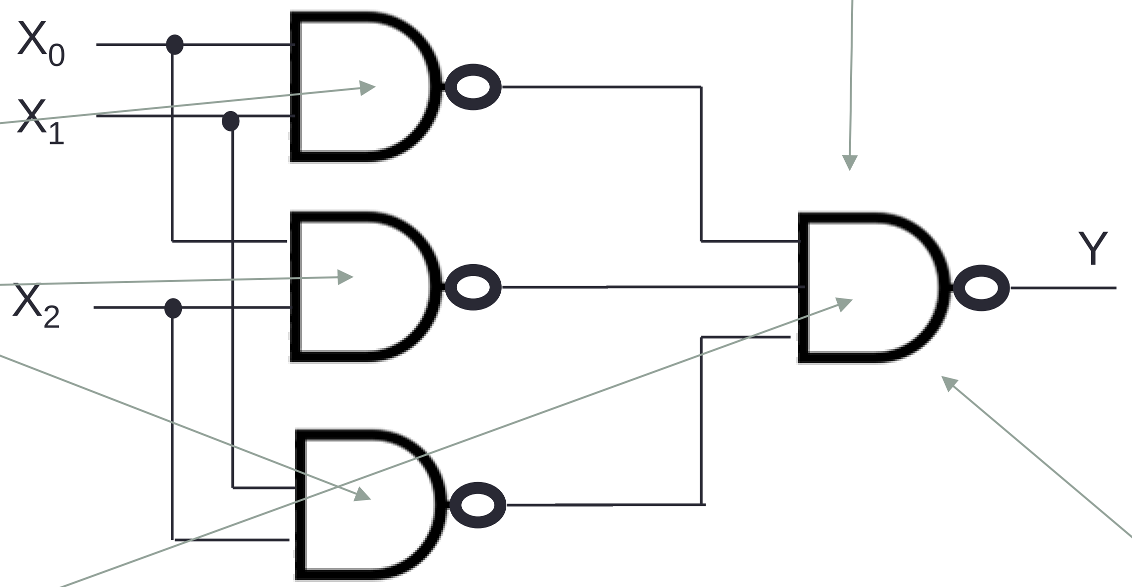
$$Y=X_1X_0+X_2X_0+X_2X_1$$

$$Y=\overline{\overline{X_1X_0}}+\overline{\overline{X_2X_0}}+\overline{\overline{X_2X_1}}$$

3 x 2NING- EI

$$Y=\overline{X_1|X_0}+\overline{X_2|X_0}+\overline{X_2|X_1}$$

De Morgani seadus $\overline{X_1\cdot X_0}=\overline{X}_1+\overline{X}_0$



Aga kui meil ei ole 3NING-EI elementi ?
Tuleb teha asendus .

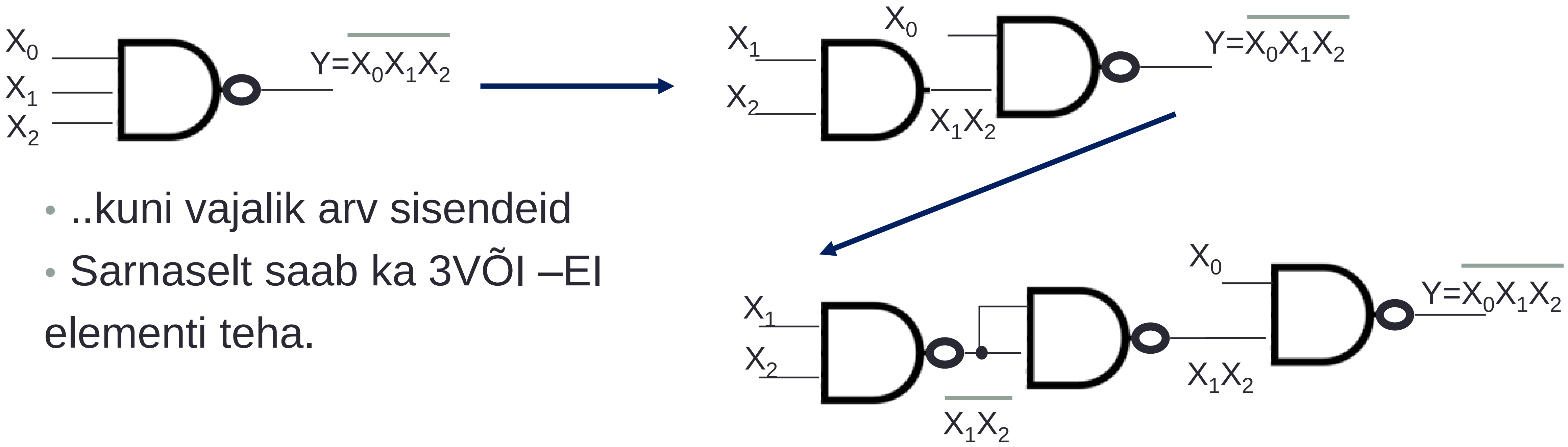
$Y=(X_1|X_0)\ |\ (X_2|X_0)\ |\ (X_2|X_1) \longrightarrow$ Vaja 3NING –EI + 3*2NING-EI

Loogikaseadmete süntees etteantud baasi alusel

- Kui loogikaelemendi (NING-EI, VÕI-EI) sisendeid jääb üle:

Kui LE kõik sisendid saavad ühe ja sama signaali, siis LE muutub inverteriks (EI tehe)

- Kui sisendeid jääb puudu – siis asendus :



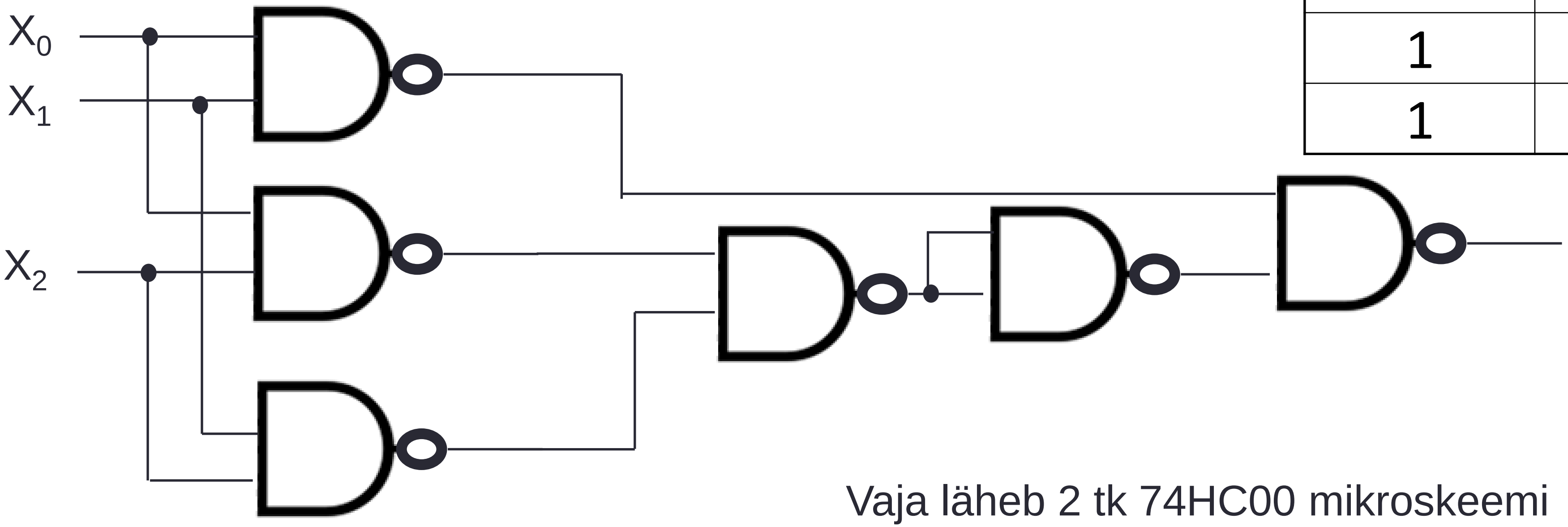
- ..kuni vajalik arv sisendeid
- Sarnaselt saab ka 3VÕI –EI

elementi teha.

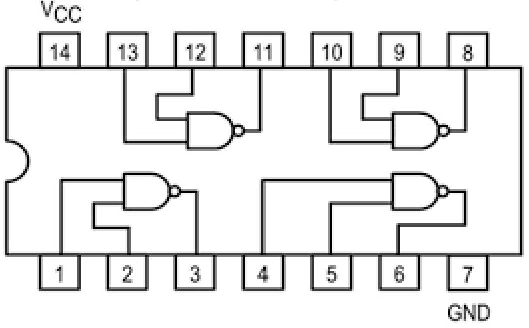
Ehk siis…..see skeem 2NING-EI elementidega

NB ! Mitte midagi ei toimu hetkega !
Tekib olekute “võidujooks” X_0 → Y tee peal 2 elementi
X_1→ Y 2 või 4 elementi , X_2 → Y 4 elementi.
Iga loogikaelement annab hilistumise .
Praktikas võib Y muutuda lühiajaliselt nulliks kui X_2=1
Ja samal hetkel muutub X_0→0 ja X_1→1
Suurel töösagedusel ,nt andmeside ,on see oluline.

X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**



$$Y=X_1X_0+X_2X_0+X_2X_1$$



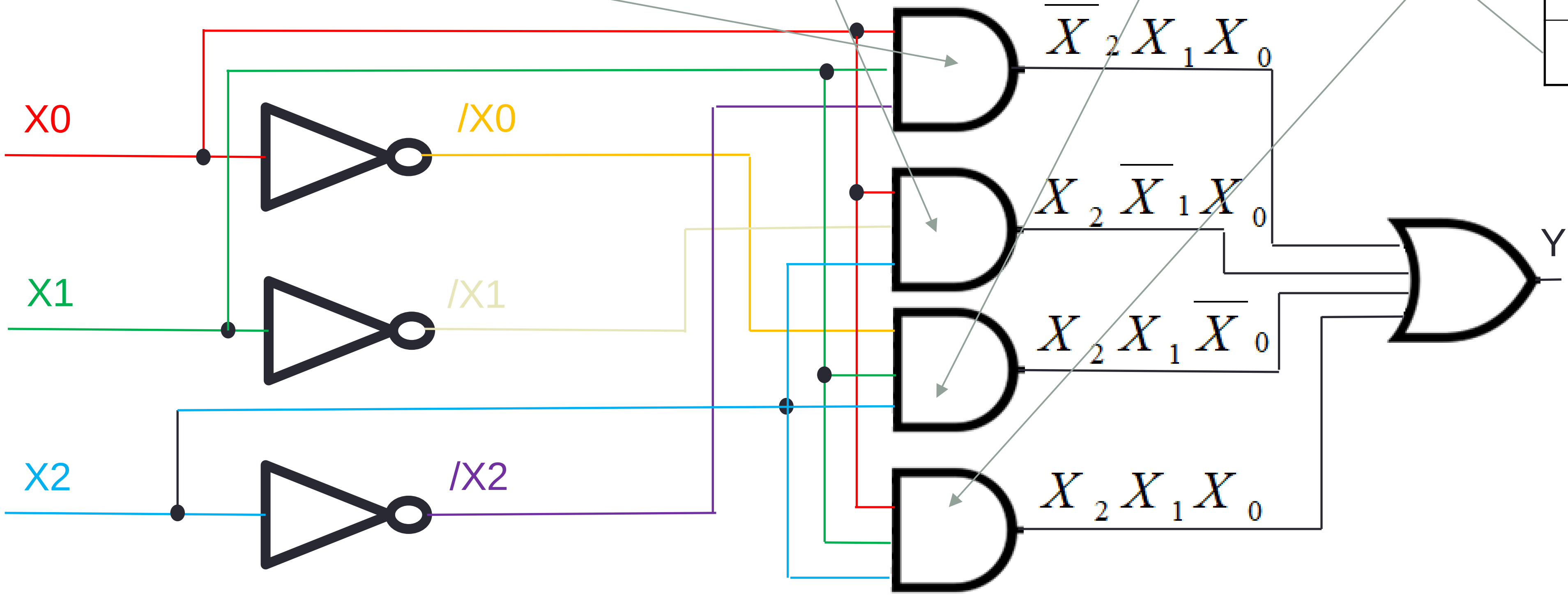
Vaja läheb 2 tk 74HC00 mikroskeemi

Loogikafunktsioonist – skeem DNV

- Sama funktsionaalsus…enne lihtsustamist
- Vaja läheb 3 tk EI , 4 tk 3NING ja 1 tk 4 OR elementi

X_2	X_1	X_0	Y
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

$$Y(X_2X_1X_0) = \overline{X}_2X_1X_0 + X_2\overline{X}_1X_0 + X_2X_1\overline{X}_0 + X_2X_1X_0$$

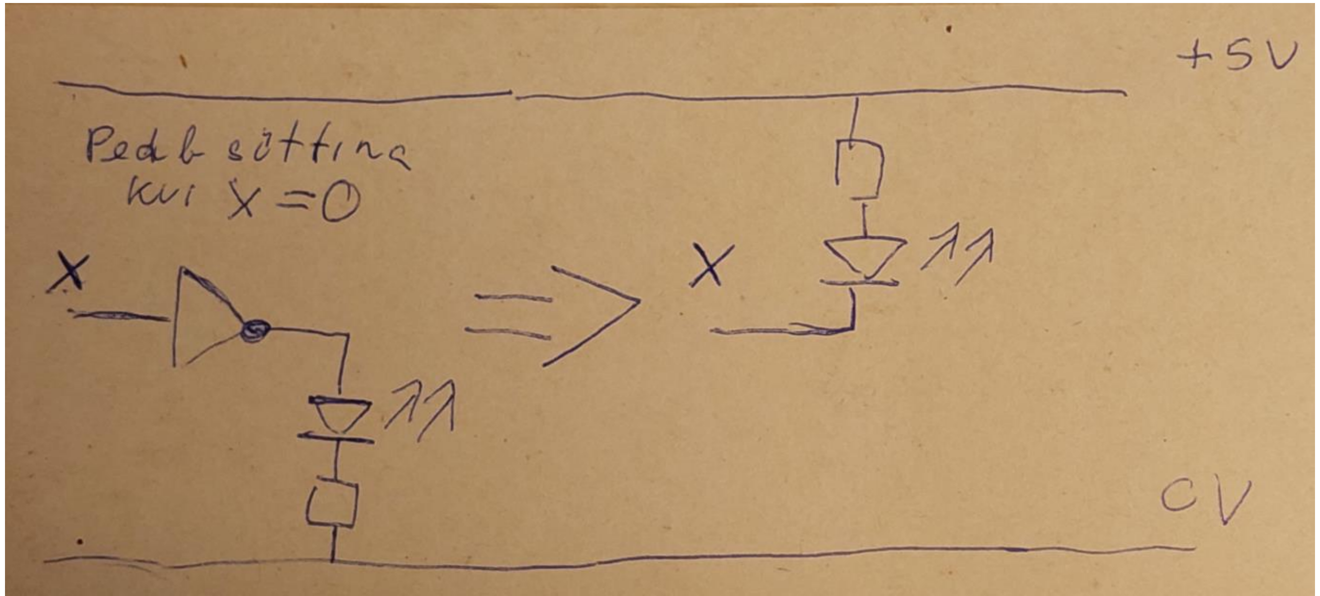


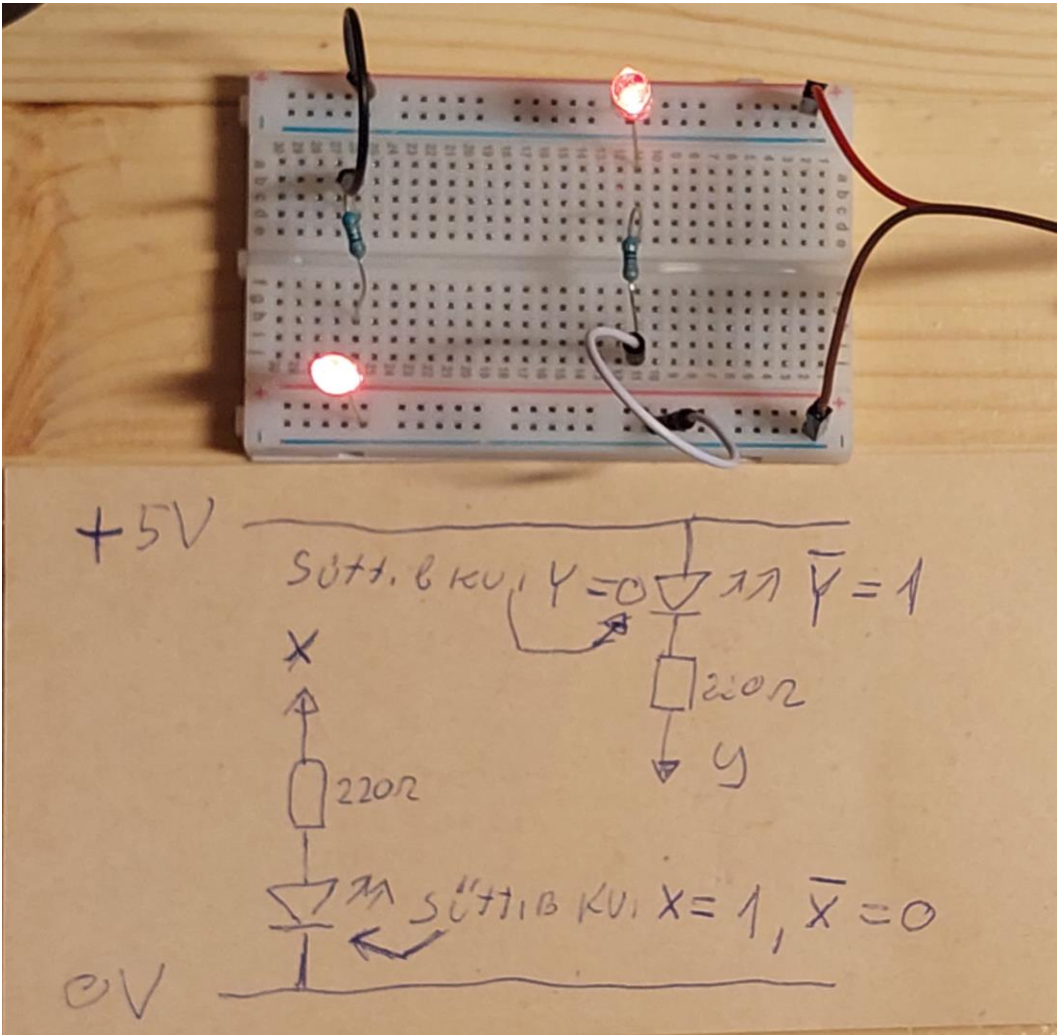
Kodutöö näide

- On siin https://youtu.be/fQ4aL3w93MY
- Lähenege loogikaelementhaaval !!!!
- Kasutage loogilist mõtlemist.
- (nt võimalik, et vahetulemusi saab grupeerida)
- Äkki on mõistlik negatiivne loogika ?
- Internet on loogikaskeemide generaatoreid täis, aga kas nad annavad mõistliku,lihtsustatud tulemuse ? (loomulikult ei keela neid kasutada)
- Proovige tulemus enne kokkupanekut simulaatoris (nt falstad.com) läbi .
- Kui välja ei tule – küsige (aga siis õppejõud soovib juurde konstruktiivset juttu) ja ta pole selgeltnägija, et vaatab juhtmerägastikule peale ja teab, mis on valesti.
- NB ! Kaitsmisel võib paha õppejõud näiteks küsida – aga milleks seda juhet on siin vaja ?

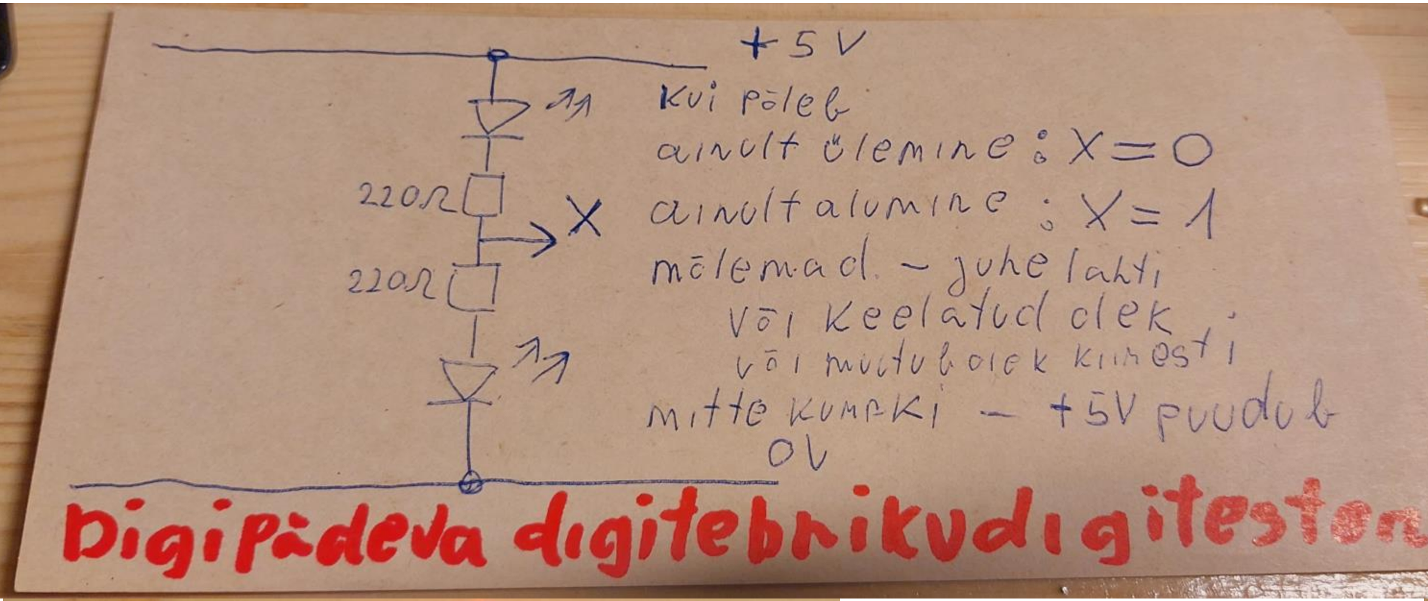
Praktilisi näpunäiteid (ka mujal kasulikud)

- Tehke endale loogikaskeemi “proovik”, kasutades valgusdioodi ja takistit.
- Inverteri (eituse) saab teha nii NING-EI kui ka VÕI-EI elemendist , ühendades sisendid kokku.
- Kui viimane element on EI , saab sellest vabaneda, tehes väljundi “negatiivsele loogikale”





Digi(pä)edeva digitehniku digitester



Kui põleb
ainult ülemine: X=0
ainult alumine: X=1
mõlemad – juhe lahti
või keelatud olek
või muutub olek kiiresti
mitte kumbki – +5V puudub

Digipädeva digitehnikudigitester