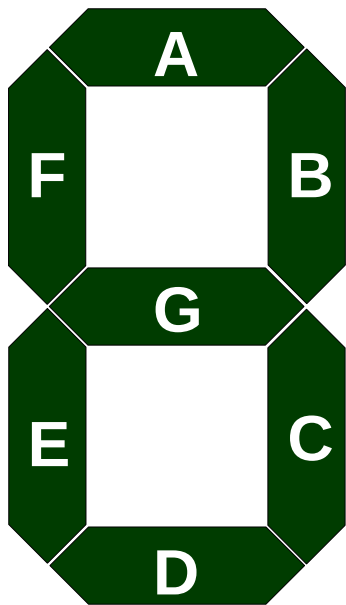


IAX0043

Segmentindikaatori juhtimine K2020

Assistent Elmet Orasson
elmet.orasson@ttu.ee

Segmentindikaator

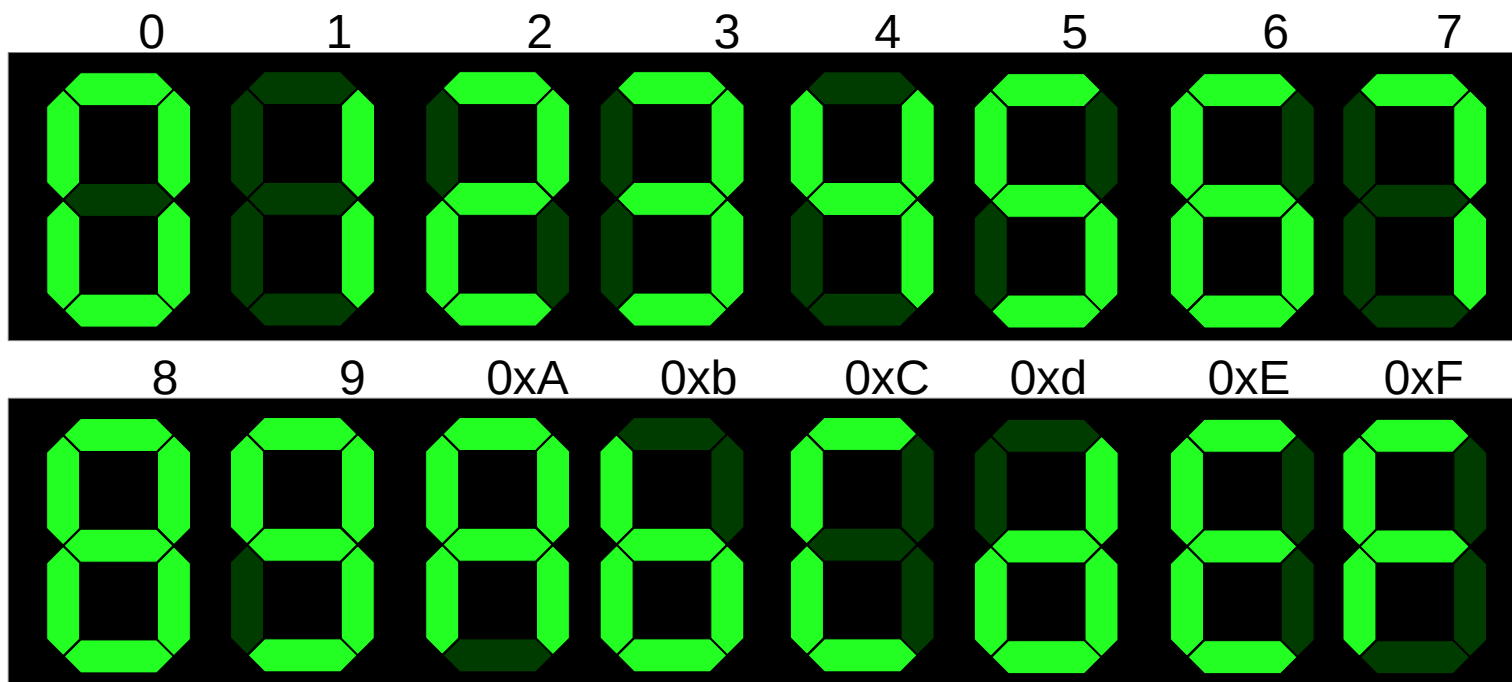


... on üks elektroonikakomponente mis on mõeldud peaaesjalikult numbrite (k.a. kuueteistkümnendsüsteem) kuvamiseks ja kasutab selleks 7-t segmenti. **Antud labori raames on segmentid tähistatud nii nagu joonisel kujutatud ning juhtskeem peab saama hakkama kõikide numbritega**

0x0-0xF ehk 0-15_{dec} ehk '0000' - '1111'_{bin}

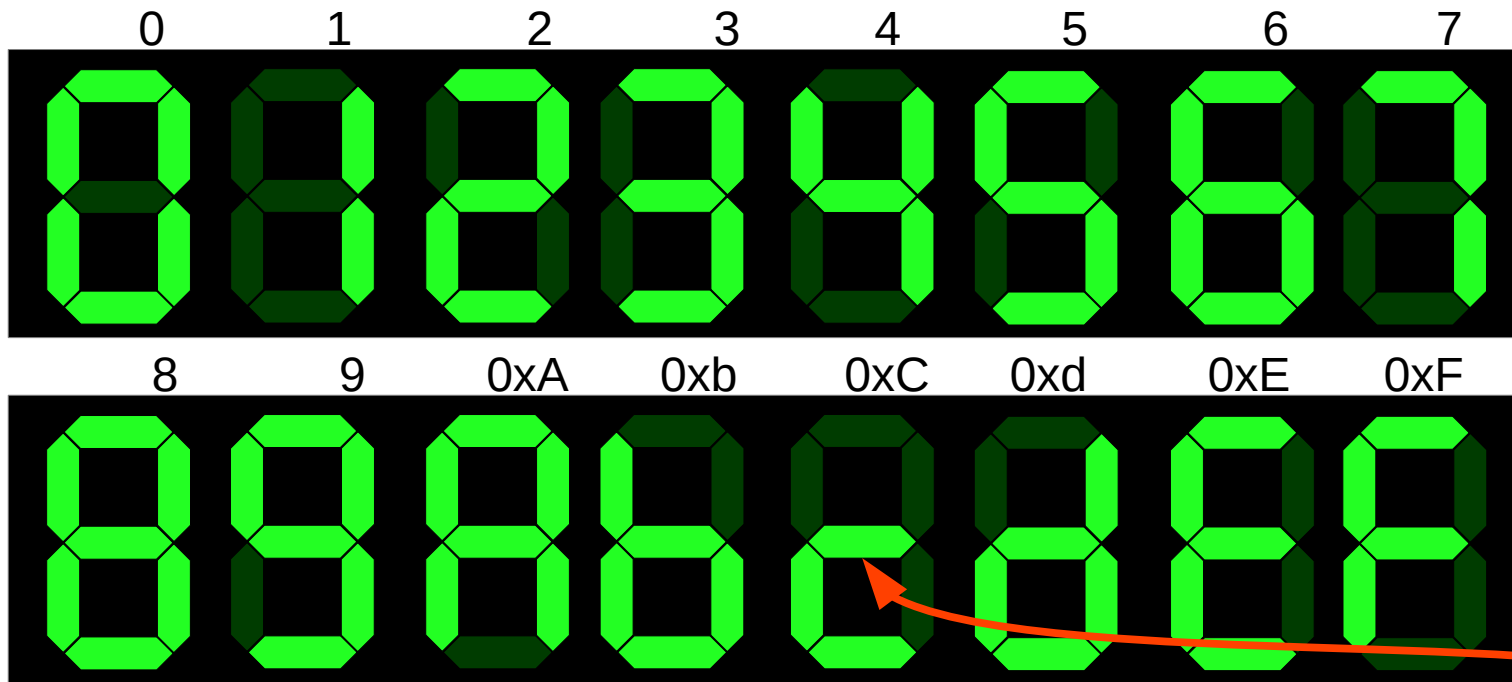
PS. Leidub indikaatoreid mis suudavad kuvada kõiki numbreid ja ladina tähestiku tähti, kuid nendel on tunduvalt enam kui 7 segmenti.

Numbrite kuvamine



aktiivsed ehk „sisselülitatud” segmendid on **helerohelised**,
so. neid juhtiva skeemi väljund on signaaliväärtusega '1'

Õppejõu näitevariant



Siin näites on kasutusel väike 'c'. Leiame tõeväärtustabeli G segmendi tarvis

Tõeväärtustabel

Kõige lihtsam on alustada tõeväärtustabelist. Kõrvalolev näide on **G** segmendi juhtimiseks, kus $0x\mathbf{C}$ asemel kuvatakse $0x\mathbf{c}$.

Digitaalelektronikas on levinud signaalide tähistamine kujul, kus muutujad (kahendarvu järgud) on seotud järgukaaludega selliselt:

muutuja/signaal $a_3 \dots a_0$
kahe aste $2^3 \dots 2^0$
järgukaal 8 4 2 1



$\#_{10}$ $\#_{16}$	$a_3 a_2 a_1 a_0$	Y
0	0000	0
1	0001	0
2	0010	1
3	0011	1
4	0100	1
5	0101	1
6	0110	1
7	0111	0
8	1000	1
9	1001	1
10	$0xA$	1
11	$0xB$	1
12	$0xC$	1
13	$0xD$	1
14	$0xE$	1
15	$0xF$	1

Karnaugh kaart

$a_3 a_2$ \ $a_1 a_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	1	1	1	1
10	1	1	1	1

Tõeväärtustabeli alusel on juba kerge Karnaugh kaarti täita ning vea kahtlustamisel neid omavahel võrrelda.

NB! Märka muutujate järjestust! See on täpselt samasugune nagu oli tõeväärtustabelis.

Disjunktivne normaalkuju (DNK)

Üldreeglid:

- Valitavad kontuurid peavad olema suurusega 2^n , 4x4 tabeli puhul $n = \{0...4\}$
- DNK puhul leiame kontuurid '1' väärtustele.
- Määramatustele omistatakse teda katva kontuuri väärtus.

DNK leidmine

$a_1 a_0$	00	01	11	10
$a_3 a_2$				
00	0	0	1	1
01	1	1	0	1
11	1	1	1	1
10	1	1	1	1

Kontuurile vastav term: $a_2 \wedge \bar{a}_1$

Märgime ühe kontuuri ja leiame teda esindava elementaarkonjunktsiooni.

DNK leidmine

$a_1 a_0$	00	01	11	10
$a_3 a_2$				
00	0	0	1	1
01	1	1	0	1
11	1	1	1	1
10	1	1	1	1

kontuur võib
laieneda üle
kaardi servade

Kontuurile vastav term (elementaarkonjunktsioon): $\bar{a}_2 \wedge a_1$

MDNK loogikaavaldis $Y = a_3 \vee (a_2 \wedge \bar{a}_1) \vee (\bar{a}_2 \wedge a_1) \vee (a_1 \wedge \bar{a}_0)$

Konjunktivne normaalkuju (KNK)

KNK leidmine toimub peegelpildis DNK leidmisele:

- '1' asemel otsime '0' kontuure
- Inverteerimised on vastupidiselt DNK termidele.
- Term koosneb loogilise liitmise tehetest (VÕI) ning termid on omavahel seotud loogilise korrutamisega (NING).

KNK leidmine

$a_1 a_0$	00	01	11	10
$a_3 a_2$				
00	0	0	1	1
01	1	1	0	1
11	1	1	1	1
10	1	1	1	1

KNK annab antud juhul lihtsama avaldisega.

2 termi, tähistatud kui roheline ja punane.

Roheline term: $a_3 \vee a_2 \vee a_1$

Punane term: $a_3 \vee \bar{a}_2 \vee \bar{a}_1 \vee \bar{a}_0$

MKNK: $Y = (a_3 \vee a_2 \vee a_1) \wedge (a_3 \vee \bar{a}_2 \vee \bar{a}_1 \vee \bar{a}_0)$

DeMorgan'i seadus

$$\overline{a_1 \wedge a_2} = \overline{a_1} \vee \overline{a_2}$$

$$\overline{a_1 \vee a_2} = \overline{a_1} \wedge \overline{a_2}$$

Seda reeglit läheb tarvis neil kes peavad oma skeemi teostama NAND või NOR elementidel.

- NAND-avaldisekuju saadakse DNK topeltinversiooniga (vt. allpool näide);
- NOR-avaldisekuju saadakse KNK topeltinversiooniga

$$Y = a_3 \vee (a_2 \wedge \overline{a_1}) \vee (\overline{a_2} \wedge a_1) \vee (a_1 \wedge \overline{a_0})$$

$$\overline{\overline{a_3 \vee (a_2 \wedge \overline{a_1}) \vee (\overline{a_2} \wedge a_1) \vee (a_1 \wedge \overline{a_0})}}$$

$$\overline{\overline{a_3} \wedge (\overline{a_2 \wedge \overline{a_1}}) \wedge (\overline{\overline{a_2} \wedge a_1}) \wedge (\overline{a_1 \wedge \overline{a_0}})}$$

Skeemi süntees I

Analüüsime eespool näite MDNK valemit:

$$Y = a_3 \vee (a_2 \wedge \bar{a}_1) \vee (\bar{a}_2 \wedge a_1) \vee (a_1 \wedge \bar{a}_0)$$

Diagram illustrating the implementation of the MDNK formula using logic gates. The formula is $Y = a_3 \vee (a_2 \wedge \bar{a}_1) \vee (\bar{a}_2 \wedge a_1) \vee (a_1 \wedge \bar{a}_0)$. The diagram shows three 2-input AND gates in the middle row, each implementing one of the three product terms. The outputs of these three AND gates are connected to a single 4-input OR gate in the bottom row. The label "2AND" is placed under each of the three AND gates, and "4OR" is placed under the OR gate.

Vajalike komponentide loend on seega järgmine:

- 2x INV
- 3x 2AND
- 1x 4OR

Skeemi süntees II

Analüüsimise MDNK ja DeMorgan'i seaduse rakendamisest saadud valemit:

$$\overline{\overline{a_3}} \wedge \overline{(a_2 \wedge \overline{a_1})} \wedge \overline{(\overline{a_2} \wedge a_1)} \wedge \overline{(a_1 \wedge \overline{a_0})}$$

Diagram illustrating the implementation of the Boolean expression using NAND gates:

- The expression is $\overline{\overline{a_3}} \wedge \overline{(a_2 \wedge \overline{a_1})} \wedge \overline{(\overline{a_2} \wedge a_1)} \wedge \overline{(a_1 \wedge \overline{a_0})}$.
- The expression is broken down into three 2NAND components: $\overline{(a_2 \wedge \overline{a_1})}$, $\overline{(\overline{a_2} \wedge a_1)}$, and $\overline{(a_1 \wedge \overline{a_0})}$.
- Each of these three components is implemented using a 2NAND gate.
- The three 2NAND components are then combined using a 4NAND gate to implement the final expression.

Vajalike komponentide loend on seega järgmine:

- 7x 2NAND (4x EI ehk inverter NAND baasil + 3x 2NAND)
- 1x 4NAND

Skeemi süntees III

Analüüsimise MKNK näitest pärit valemit:

$$\underbrace{(a_3 \vee a_2 \vee a_1)}_{3\text{OR}} \wedge \underbrace{(a_3 \vee \bar{a}_2 \vee \bar{a}_1 \vee \bar{a}_0)}_{4\text{OR}}$$

$\underbrace{\hspace{15em}}_{2\text{AND}}$

Vajalike komponentide loend on seega järgmine:

- 1x 3OR
- 1x 4OR
- 1x 4AND
- 3x INV

Skeemi süntees IV

Analüüsimise MKNK näites saadud valemit, millele on juba rakendatud DeMorgan'i seadust:

$$\overline{\overline{(a_3 \vee a_2 \vee a_1)} \vee \overline{(a_3 \vee \overline{a_2} \vee \overline{a_1} \vee \overline{a_0})}}$$

3NOR

4NOR

2NOR

Vajalike komponentide loend on seega järgmine:

- 1x 3NOR
- 1x 4NOR
- 4x 2NOR (3 inverteriteks) + funktsiooni väljund

Falstad Circuit Simulator

Otsime üles ja käivitame Falstad Circuit Simulatori (<http://www.falstad.com/circuit>).
Soovitavalt valime rakenduse alt lingi „**Full screen**”. Seejärel tekitame endale tühja töölehe:
Circuits -> Blank Circuit

Enamik meile vajalikust on kättesaadav alljärgnevate kiirvalikutega (shortcuts):

Funktsioon	Kiirvalik	Funktsioon	Kiirvalik	Funktsioon	Kiirvalik
Input	i	INV	1	Nihuta kõik	Alt+drag
Output	o				
NAND	@	AND	2	Nihuta rida	Alt+s+drag
NOR	#	OR	3	Nihuta veergu	Alt+m+drag
XOR	4	ühendused	w	Zoom in	+
Text/Label	t	(wire)		Zoom out	-

Skeemi koostamine I

Esimesena paigutame ära loogikaelemendid, arvestades et sisendite-väljundite ning ühenduste tarvis jääks piisavalt vaba ruumi. Joondame elemendid või tõstame vajadusel ringi. Loogikaelemendid ei tohiks olla üksteisele liiga lähedal või osaliselt üksteist katta (skeemi loetavus!)

Seejärel paneme paika sisendid-väljundid (arvestusega et kõrgeim sisendi järk jääb ülespoole, madalaim alla):

Draw -> Logic gates, Input and Output, vastavalt shortcut **'i'**nput ja **'o'**utput ja märgistame need kohealt!

Draw-> Outputs and Labels -> Text või shortcut **'t'**

Sellega on meil skeemi põhiplaan maha pandud ja alles nüüd võime hakata komponente ühendama.

Skeemi koostamine II

- Lisame väljunsignaali jälgimiseks 7-segmendilise indikaatori. Otsime üles oma ülesandevariandile vastava segmendi sisendi ja ühendame sinna oma skeemi väljundi. (**Draw -> Digital chips -> Add 7 segment LED**)
- Lisame loenduri (counter) ning loenduri sünkroniseerimiseks CLK (clock) sisendi (Inputs and Sources). Loenduri lähedusse paigutame veel '7 segment decoder' ning veel ühe '7 segment LED' elemendi. Sedasi saame jälgida skeemi sisendite olekuid.
- Kui näete veateadet 'Singular matrix' ja Falstad simulaator keeldub skeemi simuleerimast siis tuleb üle vaadata kas skeemi väljundi (output objekt) asemel pole sinna ühendatud sisend (input objekt).

Skeemi koostamine III

Teisel sammul paneme paika loogikaelementide vahelised ühendused.

Ühenduste (wire) tekitamine käib lõikude (ehk siis juhtmejuppide) kaupa ja peab jälgima et juhtmeotsad oleksid täpselt kokku ühendatud. Pisut nihkes ühendused ei pruugi simulaatori mõistes tegelikult ühendatud olla, kuigi peale vaadates paistavad OK!

Kui kusagil juhtmete ristumiskohas paistab punane täpp siis seal kohas puudutab ühe juhtme ots teist juhett kuid toimivat ühenduspunkti pole.

NB! Hea tava ütleb et juhtmed on vaid vertikaalsed või horisontaalsed. Juhtmed peavad olema viisakalt teistest skeemielementidest eemal (st. ei riiva ega läbi neid). Paralleelselt veetud juhtmed võiks olla ühesuguste vahedega.

Skeemi salvestamine ja laadimine

Falstad pakub selleks mitmeid võimalusi (File menüü alt leitavad) :

- Import/Export as Text – loeb/väljastab skeemikirjelduse tekstina
- Export as Link – kogu skeemikirjeldus kodeeritakse URL-i
- Dropbox
- Salvestab/loeb failist

<http://www.falstad.com/circuit>