TALLINNA TEHNIKAÜLIKOOL
INFOTEHNOLOOGIA TEADUSKOND
IT SÜSTEEMIDE ARENDUS

Arti Zirk 176119IDDR

LOOGIKAFUNKTSIOONIDE SÜSTEEM

Kodutöö 1

Tallinn 2019

Autorideklaratsioon

Kinnitan, et olen koostanud antud kodutöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor:	Arti Zirk
Kuupäev:	Oktoober 31, 2019

Sisukord

1. Lähteülesande genereerimine

Loogikaülesanne luuakse matrikli 176119IDDR järgi.

fn	ühed	määramatused
1:	`20CE0223`	`AEF560B`
2:	`19345B97`	`866C932`
3:	`99B1AD47`	`333B39C2`
4:	`171022C3`	`7B00B96`

Table 1. Ühtede ja määramatuste piirkonnad funktsioonides

Tabeli 1 abil saadakse järgmine funktsioonide süsteem:

$$
\begin{aligned}
f1(x_1, x_2, x_3, x_4) &= \sum(0, 2, 3, 12, 14)_1(5, 6, 10, 11, 15)_{-} \\
f2(x_1, x_2, x_3, x_4) &= \sum(1, 3, 4, 5, 7, 9, 11)_1(2, 6, 8, 12)_{-} \\
f3(x_1, x_2, x_3, x_4) &= \sum(1, 4, 7, 9, 10, 11, 13)_1(2, 3, 12)_{-} \\
f4(x_1, x_2, x_3, x_4) &= \sum(0, 1, 2, 3, 7, 12)_1(6, 9, 11)_{-}
\end{aligned}
$$

Samaväärne tõeväärtustabel on kuvatud tabelis 2

	x_1	x_2	x_3	x_4	$f1$	$f2$	$f3$	$f4$
0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	1	1	1
2	0	0	1	0	1	-	-	1
3	0	0	1	1	1	1	-	1
4	0	1	0	0	0	1	1	0
5	0	1	0	1	-	1	0	0
6	0	1	1	0	-	-	0	-
7	0	1	1	1	0	1	1	1
8	1	0	0	0	0	-	0	0
9	1	0	0	1	0	1	1	-
A	1	0	1	0	-	0	1	0
B	1	0	1	1	-	1	1	-
C	1	1	0	0	1	-	-	1
D	1	1	0	1	0	0	1	0
E	1	1	1	0	1	0	0	0
F	1	1	1	1	-	0	0	0

Table 2. Funktsioonide süsteem tõeväärtustabeli kujul

2. Minimeerimine

Minimeerimine on tehtud programmi `espresso`[1] abil. Sisendiks on tabelis 2 toodud tõeväärtus tabel. Argumendi `-Dopoall` kasutamisel olen välja valinud kaks huvitavamat lahendust erinevate faasidega.

Kodutöö ülesande kirjelduses on öeldud, et tuleks optimeerida pindala ning kiirust. Lisaks on piiratud, et loogika elementidel saab olla kuni 3 sisendit. Ehk siis `espresso -Dopoall` väljundis on tähtis jälgida implikantide arvu (c) ning sisendite ja väljundite koguarvu (tot).

Listing 2.1. -Dopoall faas 0000

```
# phase is ---- 0000
# ESPRESSO      Time was 0.00 sec, cost is c=9(0) in=26 out=16 tot=42
```

Faasi 0000 implikantide arv on c=9, teiste faaside korral on see arv suurem.

Kogu `espresso -Dopoall` leiab lisade all olevast listingust nr1.

Listing 2.2. -o eqntott faas 0000

```
y1 = (!x3&x4) | (!x1&x2) | (x1&!x2&!x4);
y2 = (!x2&!x3&!x4) | (x1&x2&x4) | (x1&!x2&!x4) | (x1&x2&x3);
y3 = (x2&x3&!x4) | (!x1&x2&!x3&x4) | (!x2&!x3&!x4) | (x1&x2&x3);
y4 = (!x1&x2&!x3&x4) | (!x1&!x3&!x4) | (x1&x2&x4) | (x1&!x2&!x4) |
     (x1&x2&x3);
```

Kui vaadata `-eqntott` argumendi väljundit faaasi 0000 korral siis on näha, et meil tekib väga palju üle 3 sisendiga loogika elemente. Faasi 0000 valimine edasiseks optimeerimiseks ei oleks mõtekas.

[1] `https://github.com/classabbyamp/espresso-logic`

Listing 2.3. -Dopoall faas 0101

```
# phase is ---- 0101
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=27 out=12 tot=39
```

Faasi 0101 korral on näha, et implikante on ühe võrra rohkem kuid kogu sisendite ja väljundite arv tot=39 mis on võrreldes teiste faasidega kõige väiksem.

Listing 2.4. -o eqntott faas 0101

```
y1 = (x1&!x2) | (!x3&x4) | (!x1&x2);
y2 = (!x2&x4) | (!x1&x2);
y3 = (!x1&x2&!x3&x4) | (!x1&x3&!x4) | (x1&x2&x3) | (!x2&!x3&!x4);
y4 = (x1&x2&!x3&!x4) | (!x1&x3) | (!x2&x4);
```

Vaadates `-eqntott` argumendiga saadud tulemust näeme, et üle 3 sisendiga loogika elemente on siin ka üsna vähe.

Edasiseks optimeerimiseks võtame aluseks faasi 0101.

3. Minimeerimine

3.1 Esialgne funktsioonide süsteemi skeem

Kirjutatakse välja esialgne skeem ilma elementide sisendite piiranguta ning see järel natuke lahti kirjutades vastavalt ülesande 3 sisendiga loogikaelementide piirangule.

$$
\begin{aligned}
t_1 &= x_1\,\overline{x_2}\\
t_2 &= \overline{x_3}\,x_4\\
t_3 &= \overline{x_1}\,x_2\\
t_4 &= \overline{x_2}\,x_4\\
t_5 &= \overline{x_1}\,x_2\\
t_6 &= \overline{x_1}\,x_2\,\overline{x_3}\,x_4\\
t_7 &= \overline{x_1}\,x_3\,\overline{x_4}\\
t_8 &= x_1x_2x_3\\
t_9 &= \overline{x_2}\,\overline{x_3}\,\overline{x_4}\\
t_{10} &= x_1x_2\,\overline{x_3}\,\overline{x_4}\\
t_{11} &= \overline{x_1}\,x_3\\
t_{12} &= \overline{x_2}\,x_4\\
y_1 &= \overline{t_1|t_2|t_3}\\
y_2 &= t_4|t_5\\
y_3 &= \overline{t_6|t_7|t_8|t_9}\\
y_4 &= t_{10}|t_{11}|t_{12}
\end{aligned}
$$

$$
\begin{aligned}
t_1 &= x_1\,\overline{x_2}\\
t_2 &= \overline{x_3}\,x_4\\
t_3 &= \overline{x_1}\,x_2 && [= t_5]\\
t_4 &= \overline{x_2}\,x_4\\
t_5 &= \overline{x_1}\,x_2 && [= t_3]\\
t_{61} &= \overline{x_1}\,\overline{x_3}\\
t_6 &= t_{61}x_2x_4\\
t_7 &= \overline{x_1}\,x_3\,\overline{x_4}\\
t_8 &= x_1x_2x_3 && [= t_{101}x_3]\\
t_9 &= \overline{x_2}\,\overline{x_3}\,\overline{x_4}\\
t_{101} &= x_1x_2\\
t_{10} &= t_{101}\,\overline{x_3}\,\overline{x_4}\\
t_{11} &= \overline{x_1}\,x_3\\
t_{12} &= \overline{x_2}\,x_4\\
y_1 &= \overline{t_1|t_2|t_3}\\
y_2 &= t_4|t_5\\
y_{31} &= t_6|t_7\\
y_3 &= \overline{y_{31}|t_8|t_9}\\
y_4 &= t_{10}|t_{11}|t_{12}
\end{aligned}
$$

3.2 Pindala ja viite analüüs

Järgmisena analüüsitakse loogikafunktsioonide süsteemi pindala ja viidet. Viite parameetrid on võetud koduse ülesande lehelt ning on toodud tabelis 3.

Element	Suurus	Viide
2-NAND	1.0	1.0
bnot	1.5	1.5
2-NOR	1.5	1.5
3-NAND	1.5	1.5
2-OR	2.0	2.0
2-AND	2.0	2.0
2-XOR	2.0	2.0
3-NOR	2.0	2.0
3-OR	2.5	2.5
3-AND	2.5	2.5
3-XOR	2.5	2.5

Table 3. Loogika elementide suurused ja viited

Implikantide järel on välja toodud realiseeritava loogika elemendi pindala/viide ning kogu viide.

$x_{1i} = \overline{x_1}$	$[1.5/1.5]$	1.5
$x_{2i} = \overline{x_2}$	$[1.5/1.5]$	1.5
$x_{3i} = \overline{x_3}$	$[1.5/1.5]$	1.5
$x_{4i} = \overline{x_4}$	$[1.5/1.5]$	1.5
$c_1 = x_{1i}x_2$	$[2.0/2.0]$	$1.5 + 2.0 = 3.5$
$c_2 = x_1x_2$	$[2.0/2.0]$	2.0
$t_1 = x_1x_{2i}$	$[2.0/2.0]$	$1.5 + 2.0 = 3.5$
$t_2 = x_{3i}x_4$	$[2.0/2.0]$	$1.5 + 2.0 = 3.5$
$t_4 = x_{2i}x_4$	$[2.0/2.0]$	$1.5 + 2.0 = 3.5$
$t_{61} = x_{1i}x_{3i}$	$[2.0/2.0]$	$1.5 + 2.0 = 3.5$
$t_6 = t_{61}x_2x_4$	$[2.5/2.5]$	$3.5 + 2.5 = 6.0$
$t_7 = x_{1i}x_3x_{4i}$	$[2.5/2.5]$	$1.5 + 2.5 = 3.5$
$t_8 = c_2x_3$	$[2.0/2.0]$	$2.0 + 2.0 = 4.0$
$t_9 = x_{2i}x_{3i}x_{4i}$	$[2.5/2.5]$	$1.5 + 2.5 = 3.5$
$t_{10} = c_2x_{3i}x_{4i}$	$[2.5/2.5]$	$2.0 + 2.5 = 4.5$
$t_{11} = x_{1i}x_3$	$[2.0/2.0]$	$1.5 + 2.0 = 3.5$
$t_{12} = x_{2i}x_4$	$[2.0/2.0]$	$1.5 + 2.0 = 3.5$
$y_{1i} = t_1 \| t_2 \| c_1$	$[2.5/2.5]$	$3.5 + 2.5 = 6.0$
$y_1 = \overline{y_{1i}}$	$[1.5/1.5]$	$6.0 + 1.5 = 7.5$
$y_2 = t_4 \| c_1$	$[2.0/2.0]$	$3.5 + 2.0 = 5.5$
$y_{31} = t_6 \| t_7$	$[2.0/2.0]$	$6.0 + 2.0 = 8.0$
$y_{3i} = y_{31} \| t_8 \| t_9$	$[2.5/2.5]$	$8.0 + 2.5 = 10.5$
$y_3 = \overline{y_{3i}}$	$[1.5/1.5]$	$10.5 + 1.5 = 12.0$
$y_4 = t_{10} \| t_{11} \| t_{12}$	$[2.5/2.5]$	$4.5 + 2.5 = 7.0$

Elemendid: 6 x NOT, 9 x 2-AND, 4 x 3-AND, 2 x 2-OR, 3 x 3-OR.

Kokku: 24 elementi, suurus 46, kriitiline tee 12.

3.3 Ühiste alamavaldiste otsimine

$$\begin{array}{ll}
y_1 = (x_1\,\overline{x_2})|(\overline{x_3}\,x_4)|(\overline{x_1}\,x_2) & \\
/(\overline{x_3}\,x_4) \rightarrow (x_1\,\overline{x_2})|(\overline{x_1}\,x_2) & = x_1 \oplus x_2
\end{array}$$

$$y_2 = (\overline{x_2}\,x_4)|(\overline{x_1}\,x_2)$$

$$\begin{array}{ll}
y_3 = (\overline{x_1}\,x_2\,\overline{x_3}\,x_4)|(\overline{x_1}\,x_3\,\overline{x_4})|(x_1x_2x_3)|(\overline{x_2}\,\overline{x_3}\,\overline{x_4}) & \\
/\,\overline{x_1} \rightarrow (x_2\,\overline{x_3}\,x_4)|(x_3\,\overline{x_4}) & \\
/x_2 \rightarrow (\overline{x_1}\,\overline{x_3}\,x_4)|(x_1x_3) & \\
/\,\overline{x_2} \rightarrow (\overline{x_3}\,\overline{x_4}) & (\overline{x_3}\,\overline{x_4}) = c_1 \\
/x_3 \rightarrow (\overline{x_1}\,\overline{x_4})|(x_1x_2) & (x_1x_2) = c_2 \\
/\,\overline{x_3} \rightarrow (\overline{x_1}\,x_2x_4)|(\overline{x_2}\,\overline{x_4}) & \\
/\,\overline{x_4} \rightarrow (\overline{x_1}\,x_3)|(\overline{x_2}\,\overline{x_3}) &
\end{array}$$

$$\begin{array}{ll}
y_4 = (x_1x_2\,\overline{x_3}\,\overline{x_4})|(\overline{x_1}\,x_3)|(\overline{x_2}\,x_4) & \\
/(x_1x_2) \rightarrow (\overline{x_3}\,\overline{x_4}) & = c_1 \\
/(\overline{x_3}\,\overline{x_4}) \rightarrow (x_1x_2) & = c_2
\end{array}$$

Suur enamus optimeeringuid on leitud intuitiivselt ning pole siin eraldi välja toodud.

3.4 Optimeeritud pindala ja viite analüüs

$x_{1i} = \overline{x_1}$	$[1.5/1.5]$	1.5
$x_{2i} = \overline{x_2}$	$[1.5/1.5]$	1.5
$x_{3i} = \overline{x_3}$	$[1.5/1.5]$	1.5
$x_{4i} = \overline{x_4}$	$[1.5/1.5]$	1.5

$c_1 = x_1 x_2$	$[2.0/2.0]$	2.0

$t_1 = x_1 \oplus x_2$	$[2.0/2.0]$	2.0
$t_2 = x_{3i} \overline{\wedge} x_4$	$[1.0/1.0]$	$1.5 + 1.0 = 2.5$
$y_1 = t_1 t_2$	$[2.0/2.0]$	$2.5 + 2.0 = 4.5$

$t_3 = x_{2i} \overline{\wedge} x_4$	$[1.0/1.0]$	$1.5 + 1.0 = 2.5$
$t_4 = x_{1i} \overline{\wedge} x_2$	$[1.0/1.0]$	$1.5 + 1.0 = 2.5$
$y_2 = t_3 \overline{\wedge} t_4$	$[1.0/1.0]$	$2.5 + 2.0 = 4.5$

$t_{51} = x_1 \oplus x_3$	$[2.0/2.0]$	2.0
$t_{52} = x_2 x_4$	$[2.0/2.0]$	2.0
$t_5 = t_{51} \overline{\wedge} t_{52}$	$[1.0/1.0]$	$2.0 + 1.0 = 3.0$
$t_{61} = x_1 \overline{\vee} x_4$	$[1.5/1.5]$	1.5
$t_6 = t_{61} \overline{\wedge} x_3$	$[1.0/1.0]$	$1.5 + 1.0 = 2.5$
$t_7 = c_1 \overline{\wedge} x_3$	$[1.0/1.0]$	$2.0 + 1.0 = 3.0$
$t_8 = x_2 \vert x_3 \vert x_4$	$[2.5/2.5]$	2.5
$y_{31} = t_7 t_8$	$[2.0/2.0]$	$3.0 + 2.0 = 5.0$
$y_3 = t_5 t_6 y_{31}$	$[2.0/2.0]$	$5.0 + 2.0 = 7.0$

$t_9 = x_3 \overline{\vee} x_4$	$[1.5/1.5]$	1.5
$t_{10} = c_1 \overline{\wedge} t_9$	$[2.0/2.0]$	$2.0 + 2.0 = 4.0$
$t_{11} = x_{1i} \overline{\wedge} x_3$	$[1.0/1.0]$	$1.5 + 1.0 = 2.5$
$t_{12} = x_{2i} \overline{\wedge} x_4$	$[1.0/1.0]$	$1.5 + 1.0 = 2.5$
$y_{4i} = t_{10} t_{11} t_{12}$	$[2.5/2.5]$	$4.0 + 2.5 = 6.5$
$y_4 = \overline{y_{4i}}$	$[1.5/1.5]$	$6.5 + 1.5 = 8.0$

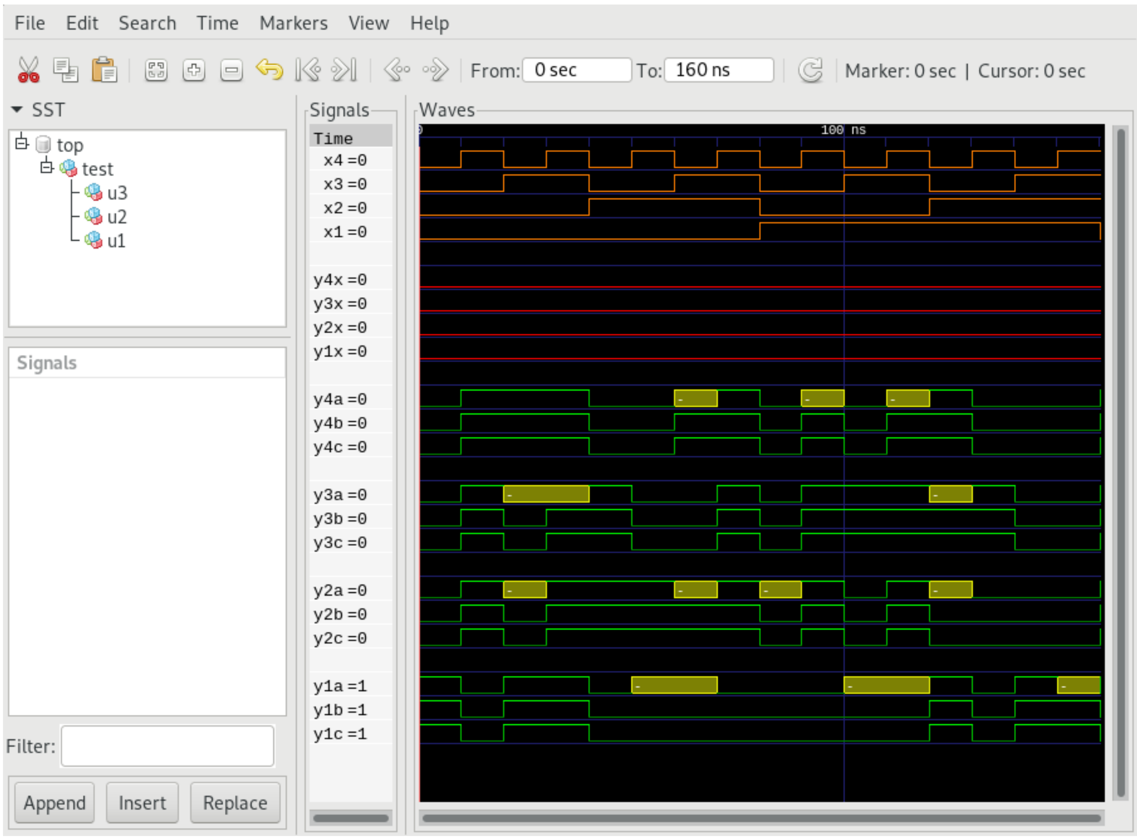
Elemendid: 5 x NOT, 4 x 2-AND, 2 x 3-AND, 1 x 3-OR, 10 x 2-NAND,

Kokku: 22 (-2) elementi, suurus 33 (-13), kriitiline tee 8 (-4).

Optimeeritud loogikasüsteemil on väiksem pindala, väiksem voolu tarve ning on kiirem.

4. Modeleerimine

Modeleerimiseks on kasutatud `ghdl` ning `gtkwave` rakendusi. VHDL koodi aluseks on kodutöö näidis projekti kood.



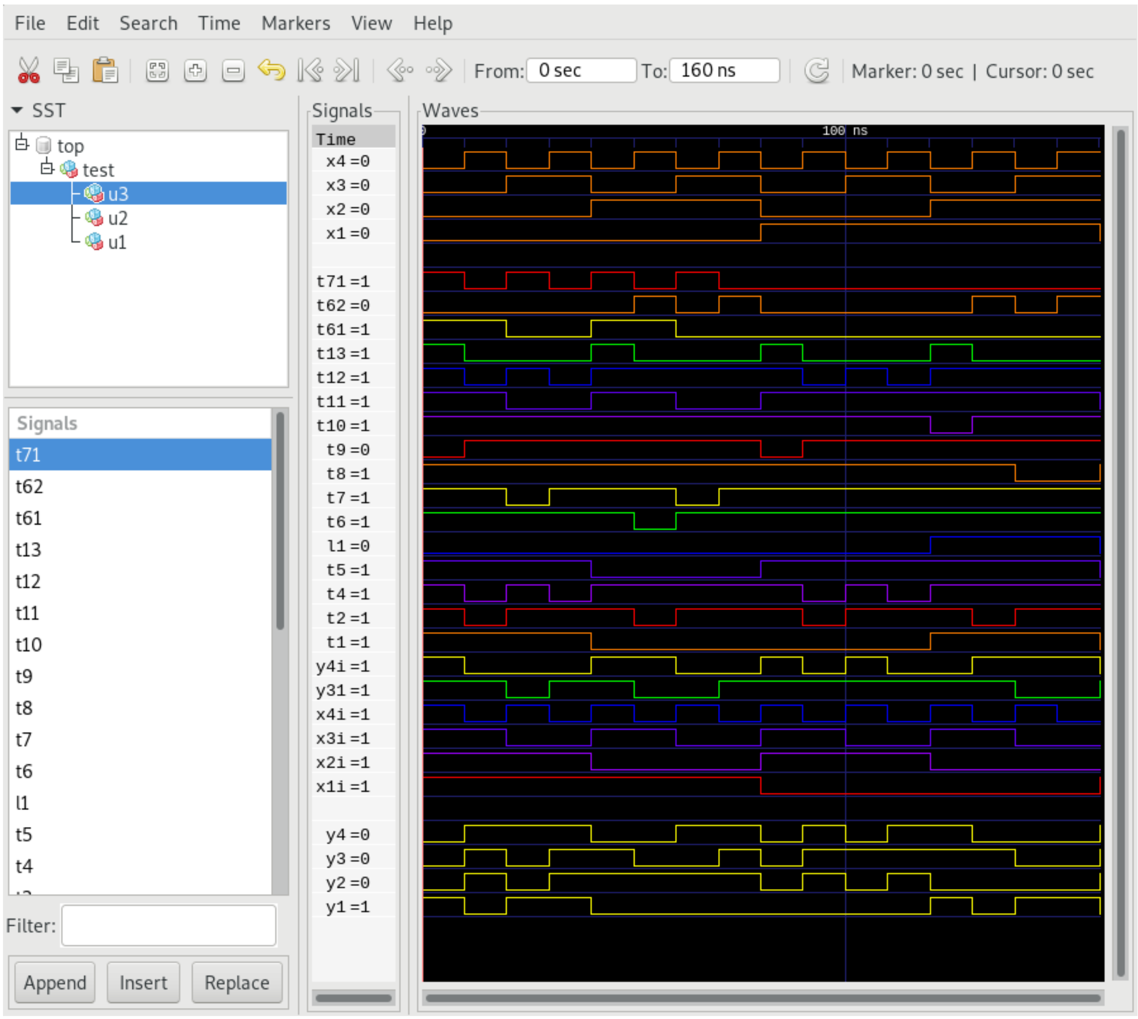
From: 0 sec To: 160 ns

Marker: 0 sec | Cursor: 0 sec

Figure 1. *Testpingi väljund*

Joonisel 1, x_1, x_2, x_3, x_4 on testpingi sisendid, y_1x, y_2x, y_3x, y_4x on testpingi väljundid ja signaalid $y\{_1,_2,_3,_4\}\{a, b, c\}$ on tabeli, espresso ja optimeeritud loogikaskeemi väljundid.

Joonis 2 kuvatud signaalid vastavad lisas toodud `opt1o.vhdl` koodile mille listing nr5 leiab lisadest. Joonis 2 kuvatud signaalid vastavad lisas toodud `opt1o.vhdl` koodile mille listing nr5 leiab lisadest.



```
Time
  x4 =0
  x3 =0
  x2 =0
  x1 =0

 t71 =1
 t62 =0
 t61 =1
 t13 =1
 t12 =1
 t11 =1
 t10 =1
  t9 =0
  t8 =1
  t7 =1
  t6 =1
  l1 =0
  t5 =1
  t4 =1
  t2 =1
  t1 =1
 y4i =1
 y31 =1
 x4i =1
 x3i =1
 x2i =1
 x1i =1

  y4 =0
  y3 =0
  y2 =0
  y1 =1
```

Figure 2. *Optimeeritud loogikaskeemi sisemised väärtused*

Lisad

Lisa 1 - espresso -Dopoall

Listing 1. Programmi espresso -Dopoall väljund

```
.i 4
.o 4
0000 1000
0001 0111
0010 1--1
0011 11-1
0100 0110
0101 -100
0110 --0-
0111 0111
1000 0-00
1001 011-
1010 -010
1011 -11-
1100 1--1
1101 0010
1110 1000
1111 -000
.e
# phase is ---- 0000
# ESPRESSO      Time was 0.00 sec, cost is c=9(0) in=26 out=16 tot=42
# phase is ---- 0001
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=30 out=15 tot=45
# phase is ---- 0010
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=30 out=18 tot=48
# phase is ---- 0011
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=28 out=15 tot=43
# phase is ---- 0100
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=26 out=15 tot=41
# phase is ---- 0101
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=27 out=12 tot=39
# phase is ---- 0110
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=27 out=17 tot=44
# phase is ---- 0111
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=26 out=14 tot=40
# phase is ---- 1000
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=30 out=15 tot=45
# phase is ---- 1001
# ESPRESSO      Time was 0.00 sec, cost is c=11(0) in=30 out=14 tot=44
# phase is ---- 1010
# ESPRESSO      Time was 0.00 sec, cost is c=11(0) in=31 out=16 tot=47
# phase is ---- 1011
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=26 out=15 tot=41
# phase is ---- 1100
# ESPRESSO      Time was 0.00 sec, cost is c=12(0) in=32 out=14 tot=46
# phase is ---- 1101
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=28 out=13 tot=41
# phase is ---- 1110
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=27 out=15 tot=42
# phase is ---- 1111
# ESPRESSO      Time was 0.00 sec, cost is c=10(0) in=26 out=15 tot=41
```

Lisa 1 - VHDL kood

Listing 2. f_system.vhd

```
library IEEE; use IEEE.std_logic_1164.all;
entity f_system is
  port ( x1, x2, x3, x4: in std_logic;
         y1, y2, y3, y4: out std_logic );
end entity f_system;
```

Listing 3. fs1_table.vhd

```
-------------------------------------
-- IAY0150 - Homework #1. Truth table
-------------------------------------
-- (C) Arti Zirk - 2019 - Tallinn
-------------------------------------

library IEEE; use IEEE.std_logic_1164.all;
architecture tabel of f_system is
```

```
begin
  process (x1, x2, x3, x4)
    variable in_word, out_word: std_logic_vector (3 downto 0);
  begin
    in_word := x1 & x2 & x3 & x4;
    case  in_word  is
      when "0000" => out_word := "1000";
      when "0001" => out_word := "0111";
      when "0010" => out_word := "1--1";
      when "0011" => out_word := "11-1";
      when "0100" => out_word := "0110";
      when "0101" => out_word := "-100";
      when "0110" => out_word := "--0-";
      when "0111" => out_word := "0111";
      when "1000" => out_word := "0-00";
      when "1001" => out_word := "011-";
      when "1010" => out_word := "-010";
      when "1011" => out_word := "-11-";
      when "1100" => out_word := "1--1";
      when "1101" => out_word := "0010";
      when "1110" => out_word := "1000";
      when "1111" => out_word := "-000";
      when others => out_word := "----";
    end case;
    y1 <= out_word(3);    y2 <= out_word(2);
    y3 <= out_word(1);    y4 <= out_word(0);
  end process;
end architecture tabel;
```

Listing 4. fs1_espr.vhd

```
-----------------------------------------
-- IAY0150 - Homework #1. Espresso result
-----------------------------------------
-- (C) Arti Zirk - 2019 - Tallinn
-----------------------------------------

library IEEE; use IEEE.std_logic_1164.all;
architecture espresso of f_system is
begin
  y1 <= not ((x1 and not x2) or (not x3 and x4) or
        (not x1 and x2));

  y2 <= (not x2 and x4) or  (not x1 and x2);
  y3 <= not ((not x1 and x2 and not x3 and x4) or
       (not x1 and x3 and not x4) or
       (x1 and x2 and x3) or
       (not x2 and not x3 and not x4));

  y4 <= (x1 and x2 and not x3 and not x4) or
        (not x1 and x3) or (not x2 and x4);
end architecture espresso;
```

Listing 5. fs1_opt1o.vhd

```
-----------------------------------------
-- IAY0150 - Homework #1.
-----------------------------------------
-- (C) Arti Zirk - 2019 - Tallinn
-----------------------------------------

library IEEE; use IEEE.std_logic_1164.all;
architecture opti1 of f_system is
  signal x1i, x2i, x3i, x4i, y31, y4i, t1: std_logic;
  signal t6, t7, t8, t9, t10, t11, t12: std_logic;
  signal t2, t3, t4, t5, l1, t13, t61, t62, t71: std_logic;
begin
  x1i <= not x1;
  x2i <= not x2;
  x3i <= not x3;
  x4i <= not x4;

  l1 <= x1 and x2;

  t1 <= not (x1 xor x2);
  t2 <= x3i nand x4;
  y1 <= t1 and t2;

  t4 <= x2i nand x4;
  t5 <= x1i nand x2;
  y2 <= t4 nand t5;

  t61 <= x1 nor x3;
  t62 <= x2 and x4;
  t6  <= t61 nand t62;
  t71 <= x1 nor x4;
  t7 <= t71 nand x3;
  t8 <= l1 nand x3;
  t9 <= x2 or x3 or x4;
  y31 <= t6 and t7 and t8;
  y3 <= y31 and t9;

  t13 <= x3 nor x4;
  t10 <= l1 nand t13;
  t11 <= x1i nand x3;
  t12 <= x2i nand x4;
  y4i <= t10 and t11 and t12;
  y4 <= not y4i;
end architecture opti1;
```

Listing 6. fs1_test.vhd

```
------------------------------------------------------------------------
-- IAY0150 - Homework #1. Test bench for the example task.
------------------------------------------------------------------------
-- (C) Peeter Ellervee - 2016 - Tallinn
------------------------------------------------------------------------
library IEEE; use IEEE.std_logic_1164.all;
entity test is
end entity test;

library IEEE; use IEEE.std_logic_1164.all;
architecture bench of test is
  signal x1, x2, x3, x4: std_logic;
  signal y1a, y1b, y1c, y2a, y2b, y2c: std_logic;
  signal y3a, y3b, y3c, y4a, y4b, y4c: std_logic;
  signal y1x, y2x, y3x, y4x: std_logic;
  component f_system
    port ( x1, x2, x3, x4: in std_logic;
           y1, y2, y3, y4: out std_logic );
  end component;
  for U1: f_system use entity work.f_system(tabel);
  for U2: f_system use entity work.f_system(espresso);
  for U3: f_system use entity work.f_system(opti1);

  function compare_signals (s1, s2, s3: std_logic) return std_logic is
  begin
    if  s1='-'  then
      if  s2/=s3  then  return 'X';  end if;
    else
      if  s1/=s2 or s1/=s3  then  return 'X';  end if;
    end if;
    return '0';
  end function compare_signals;
```

```
begin
  -- Input signals (after every 10 ns)
  x1 <= ’0’ after 0 ns, ’1’ after 80 ns, ’0’ after 160 ns;
  x2 <= ’0’ after 0 ns, ’1’ after 40 ns, ’0’ after 80 ns, ’1’ after 120 ns;
  x3 <= ’0’ after 0 ns, ’1’ after 20 ns, ’0’ after 40 ns, ’1’ after 60 ns,
        ’0’ after 80 ns, ’1’ after 100 ns, ’0’ after 120 ns, ’1’ after 140 ns;
  x4 <= ’0’ after 0 ns, ’1’ after 10 ns, ’0’ after 20 ns, ’1’ after 30 ns,
        ’0’ after 40 ns, ’1’ after 50 ns, ’0’ after 60 ns, ’1’ after 70 ns,
        ’0’ after 80 ns, ’1’ after 90 ns, ’0’ after 100 ns, ’1’ after 110 ns,
        ’0’ after 120 ns, ’1’ after 130 ns, ’0’ after 140 ns, ’1’ after 150 ns;

  -- System of Boolean functions
  U1: f_system port map (x1, x2, x3, x4, y1a, y2a, y3a, y4a);
  U2: f_system port map (x1, x2, x3, x4, y1b, y2b, y3b, y4b);
  U3: f_system port map (x1, x2, x3, x4, y1c, y2c, y3c, y4c);
  --y1c<=y1b;  y2c<=y2b;  y3c<=y3b;  y4c<=y4b;
  y1x <= compare_signals (y1a, y1b, y1c);
  y2x <= compare_signals (y2a, y2b, y2c);
  y3x <= compare_signals (y3a, y3b, y3c);
  y4x <= compare_signals (y4a, y4b, y4c);
end architecture bench;
```

Listing 7. make.sh

```
#!/bin/bash
set -x -e

# uses ghdl-mcode variant
# import vhd files
ghdl -i -g *.vhd
# make unit
ghdl -m test
# run unit and export logic graph
ghdl -r test --wave=test.ghw --stop-time=200ns

# send signal to gtkwave to reload open file
gsettings set com.geda.gtkwave reload 0
```