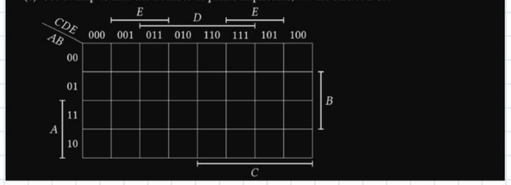
Корепин Константин M3104

Homework 3 Boolean Algebra

1. Perform the following steps:
   (a) Calculate the SHA-256 hash h of the string s = "DM Fall 2023 HW3" (without quotes, with all spaces, encoded in UTF-8). Convert hash h to a 256-bit binary string b (prepend leading zeros if necessary). Cut the binary string b into eight 32-bit slices $r_1, \ldots, r_8$, e.g. $r_2 = b_{33\ldots64}$. Xor all slices into a 32-bit string $d = r_1 \oplus \cdots \oplus r_8$. Compute $w = d \oplus$ 0x24d03294.
   Hint: last (least significant) bits of h are …01001001, last bits of d are …0001.
   (b) Draw the Karnaugh map (use a template below) for a function $f(A, B, C, D, E)$ defined by the truth table $w = (w_1 \ldots w_{32})$, where MSB corresponds to $f(0) = w_1$ and LSB to $f(1) = w_{32}$.
   (c) Use K-map to find the minimal DNF and minimal CNF for the function f.
   (d) Use K-map to find the number of prime implicants, *i.e.* the size of BCF.

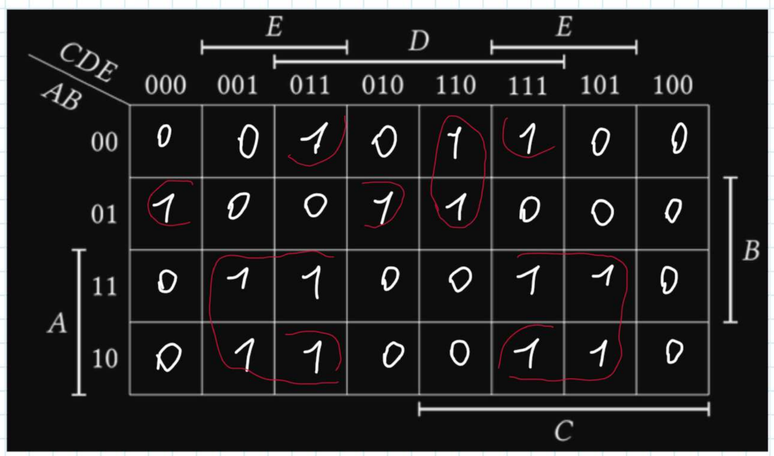


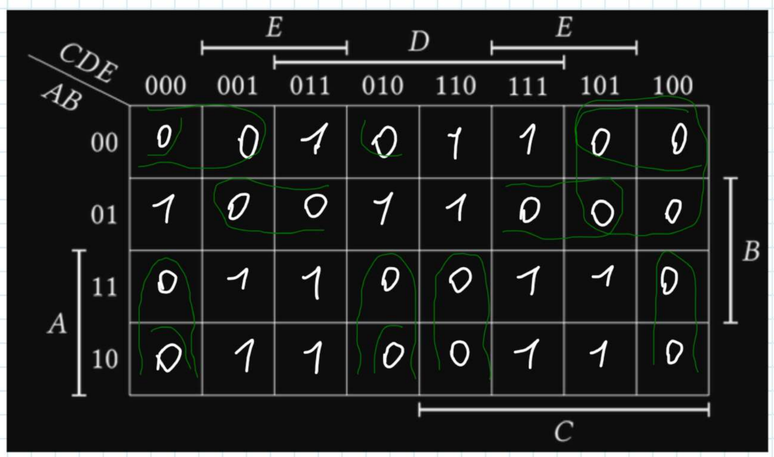
d)
s = DM Fall 2023 HW3
h = 1b772435e9a2e3fc77bef1dbab27e1b170745b0af47420f2552213d3c81cd849
b = 00011011011101110001001000001101011110100110100010111000111111110001110111101111101111000111011011010101100100111111000011011000101110000011101000101101100001010111101000111010000100000111100100101010010001000010011110100111100100000011100110110000010010011101110111000100100000110101111010011010001011100011111111000111011110111111011110001110110111010101100100111111000011011000101110000011101000101101100001010111101000111010000100000111100100101010010001000010011110100111100100000011100110110000010010011

r1 = 00011011011101110010010000110101
r2 = 11101001101000101110001111111100
r3 = 01110111101111101111000111011011
r4 = 10101011001001111110000110110001
r5 = 01110000011101000101101100001010
r6 = 11110100011101000010000011110010
r7 = 01010101001000100001001111010011
r8 = 11001000000111001101100001001001

d = 00110111011100100110011111000001
w = 00010011101000100101010101010101

b)

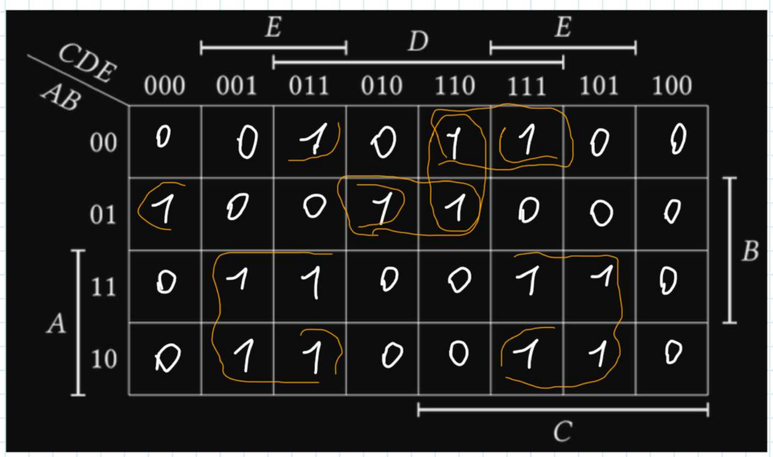




c) • mDNF = $\bar{a}b\bar{c} \vee ae \vee \bar{b}de \vee \bar{a}cd\bar{e}$

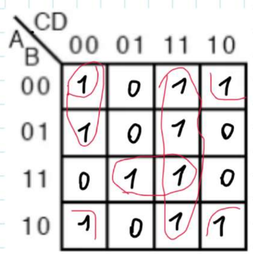
• mCNF = $(\bar{a} \vee e) \wedge (b \vee c \vee e) \wedge (a \vee \bar{b} \vee \bar{e}) \wedge (a \vee b \vee d) \wedge (a \vee \bar{c} \vee d)$

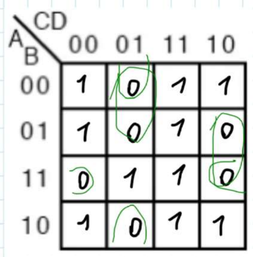
d)

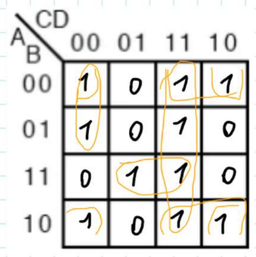


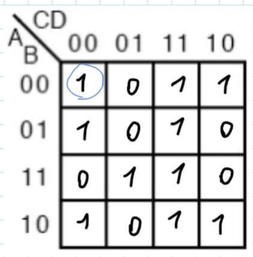
• BCF = $ae \vee \bar{b}de \vee \bar{a}b\bar{c}\bar{e} \vee \bar{a}cde \vee \bar{a}\bar{b}cd \vee \bar{a}bd\bar{e}$

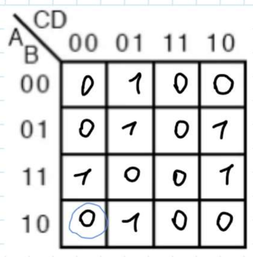
6 prime implicants

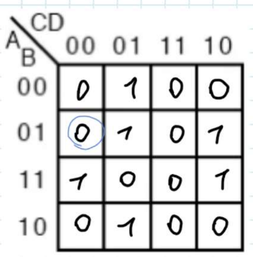


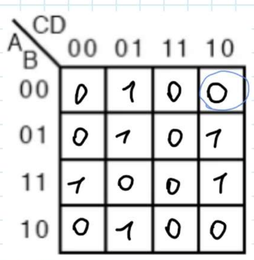


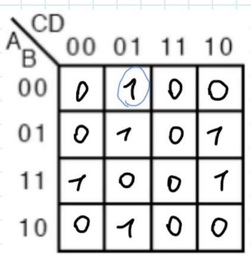


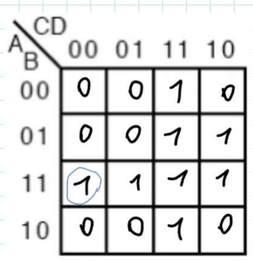


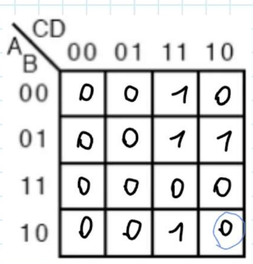


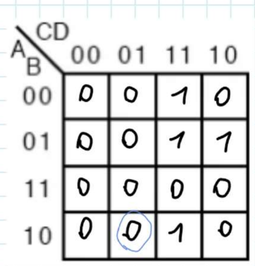


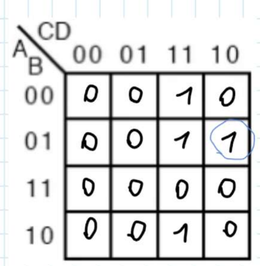


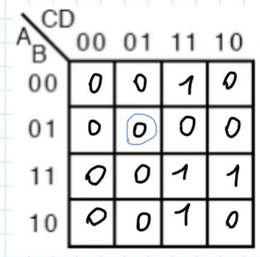


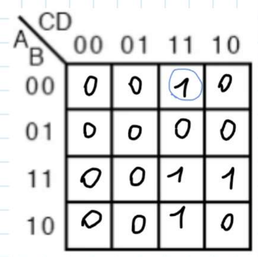


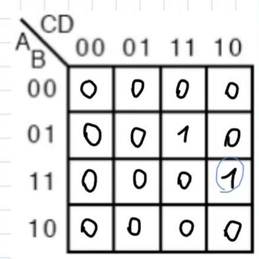


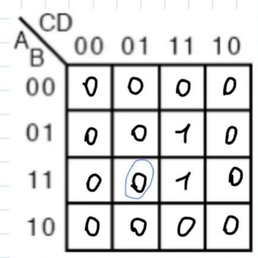


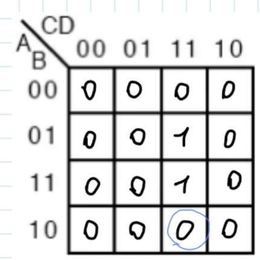


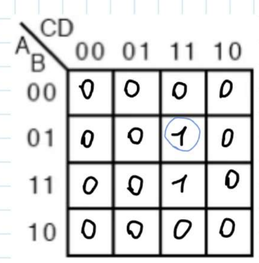


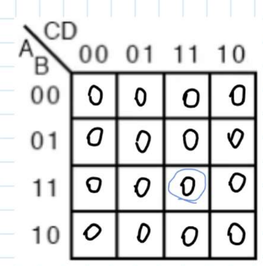


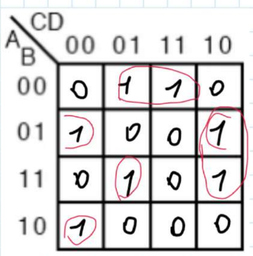


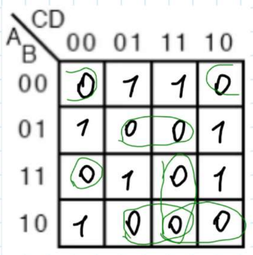


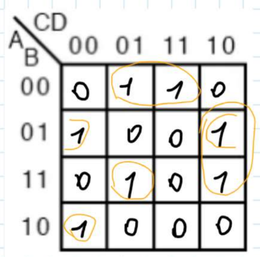


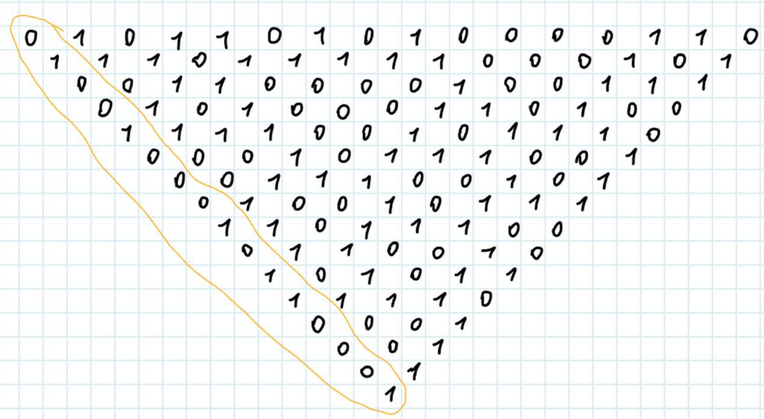












2. For each given function f_i of 4 arguments, draw the Karnaugh map and use it to find BCF, minimal DNF, and minimal CNF. Additionally, construct ANF (Zhegalkin polynomial) using either the K-map, the tabular ("triangle") method or the Pascal method — use each method at least once.

> **Note:** WolframAlpha interprets the query "*n*-th Boolean function of *k* variables" in a reverse manner. In order to employ WolframAlpha properly, manually flip the truth table beforehand, *e.g.* the correct query for $f_{10}^{(2)}$ is "5th Boolean function of 2 variables", which gives $f_{10}^{(2)} = \neg x_2$, since $\texttt{rev}(1010_2) = 0101_2 = 5_{10}$.

(a) $f_1 = f_{47541}^{(4)}$
(b) $f_2 = \sum m(1, 4, 5, 6, 8, 12, 13)$
(c) $f_3 = f_{51011}^{(4)} \oplus f_{40389}^{(4)}$
(d) $f_4 = A\overline{B}D + \overline{A}\,\overline{C}D + \overline{B}C\overline{D} + A\overline{C}D$

47541 = 1011100110110101
51011 = 1100011101000011
40389 = 1001110111000101
51011 xor 40389 = 0101101010000110

a)

AB \ CD	00	01	11	10
00	1	0	1	1
01	1	0	1	0
11	0	1	1	0
10	1	0	1	1

AB \ CD	00	01	11	10
00	1	0	1	1
01	1	0	1	0
11	0	1	1	0
10	1	0	1	1

AB \ CD	00	01	11	10
00	1	0	1	1
01	1	0	1	0
11	0	1	1	0
10	1	0	1	1

🟥 $mDNF = \bar b \bar d \lor cd \lor abd \lor \bar a \bar c \bar d$

🟩 $mCNF = (\bar a \lor \bar b \lor d) \land (\bar b \lor \bar c \lor d) \land (a \lor c \lor \bar d) \land (b \lor c \lor \bar d)$

🟧 $BCF = cd \lor \bar b c \lor \bar b \bar d \lor abd \lor \bar a \bar c \bar d$

AB \ CD	00	01	11	10
00	1	0	1	1
01	1	0	1	0
11	0	1	1	0
10	1	0	1	1

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	1	0	1
11	1	0	0	1
10	0	1	0	0

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	1	0	1
11	1	0	0	1
10	0	1	0	0

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	1	0	1
11	1	0	0	1
10	0	1	0	0

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	1	0	1
11	1	0	0	1
10	0	1	0	0

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	1	1	1	1
10	0	0	1	0

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	0	0	1	1
10	0	0	1	0

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	0	0	1	1
10	0	0	1	0

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	1
10	0	0	0	0

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	0
10	0	0	0	0

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	0
10	0	0	0	0

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	0
10	0	0	0	0

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

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

$ANF = 1 \oplus d \oplus ab \oplus bc \oplus cd \oplus abc \oplus bcd$

c)

AB \ CD	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	0	1	0	1
10	1	0	0	0

AB \ CD	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	0	1	0	1
10	1	0	0	0

🟥 $mDNF = \bar a b \bar d \lor bc\bar d \lor \bar a \bar b d \lor a \bar b \bar c \bar d \lor ab\bar c d$

🟩 $mCNF = (a \lor b \lor d) \land (\bar a \lor b \lor \bar d) \land (a \lor \bar b \lor \bar d) \land (\bar a \lor \bar b \lor c \lor d) \land (\bar a \lor b \lor \bar c) \land (\bar a \lor \bar c \lor \bar d)$

AB \ CD	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	0	1	0	1
10	1	0	0	0

🟧 $BCF = \bar a b \bar d \lor bc\bar d \lor \bar a \bar b d \lor a \bar b \bar c \bar d \lor ab\bar c d$

a	b	c	d	f	
0	0	0	0	0	1
0	0	0	1	1	d
0	0	1	0	0	c
0	0	1	1	1	cd
0	1	0	0	1	b
0	1	0	1	0	bd
0	1	1	0	1	bc
0	1	1	1	0	bcd
1	0	0	0	1	a
1	0	0	1	0	ad
1	0	1	0	0	ac
1	0	1	1	0	acd
1	1	0	0	0	ab
1	1	0	1	1	abd
1	1	1	0	1	abc
1	1	1	1	0	abcd

```
0 1 0 1 1 0 1 0 1 0 0 0 0 1 1 0
 1 1 1 0 1 1 1 1 1 0 0 0 1 0 1
  0 0 1 1 0 0 0 0 1 0 0 1 1 1
   0 1 0 1 0 0 0 1 1 0 1 0 0
    1 1 1 1 0 0 1 0 1 1 1 0
     0 0 0 1 0 1 1 1 0 0 1
      0 0 1 1 1 0 0 1 0 1
       0 1 0 0 1 0 1 1 1
        1 1 0 1 1 1 0 0
         0 1 1 0 0 1 0
          1 0 1 0 1 1
           1 1 1 1 0
            0 0 0 1
             0 0 1
              0 1
               1
```

$ANF = d \oplus b \oplus a \oplus ac \oplus acd \oplus abcd$

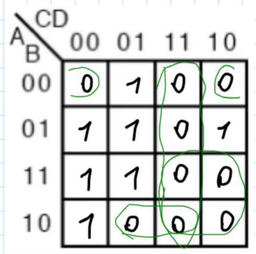
2. For each given function f_i of 4 arguments, draw the Karnaugh map and use it to find BCF, minimal DNF, and minimal CNF. Additionally, construct ANF (Zhegalkin polynomial) using either the K-map, the tabular ("triangle") method or the Pascal method — use each method at least once.

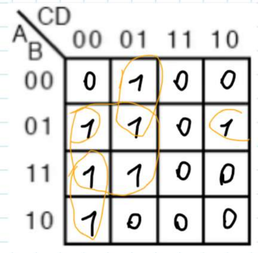
> **Note:** WolframAlpha interprets the query "n-th Boolean function of k variables" in a reverse manner. In order to employ WolframAlpha properly, manually flip the truth table beforehand, *e.g.* the correct query for $f_{10}^{(2)}$ is "5th Boolean function of 2 variables", which gives $f_{10}^{(2)} = \neg x_2$, since $\texttt{rev}(1010_2) = 0101_2 = 5_{10}$.

(a) $f_1 = f_{47541}^{(4)}$
(b) $f_2 = \sum m(1, 4, 5, 6, 8, 12, 13)$
(c) $f_3 = f_{51011}^{(4)} \oplus f_{40389}^{(4)}$
(d) $f_4 = A\overline{B}D + \overline{A}\,\overline{C}D + \overline{B}C\overline{D} + A\overline{C}D$

b)

AB \ CD	00	01	11	10
00	0	1	0	0
01	1	1	0	1
11	1	1	0	0
10	1	0	0	0





mDNF = $b\bar{c} \vee a\bar{c}\bar{d} \vee \bar{a}b\bar{d} \vee \bar{a}\bar{c}d$

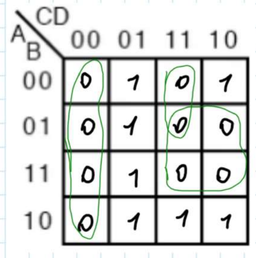
mCNF = $(\bar{a} \vee \bar{c}) \wedge (\bar{c} \vee \bar{d}) \wedge (a \vee b \vee d) \wedge (\bar{a} \vee b \vee \bar{d})$

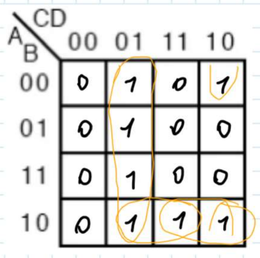
BCF = $b\bar{c} \vee a\bar{c}\bar{d} \vee \bar{a}b\bar{d} \vee \bar{a}\bar{c}d$

ANF = $d \oplus cd \oplus b \oplus bd \oplus a \oplus ac \oplus ab \oplus abcd$

d)

AB \ CD	00	01	11	10
00	0	1	0	1
01	0	1	0	0
11	0	1	0	0
10	0	1	1	1



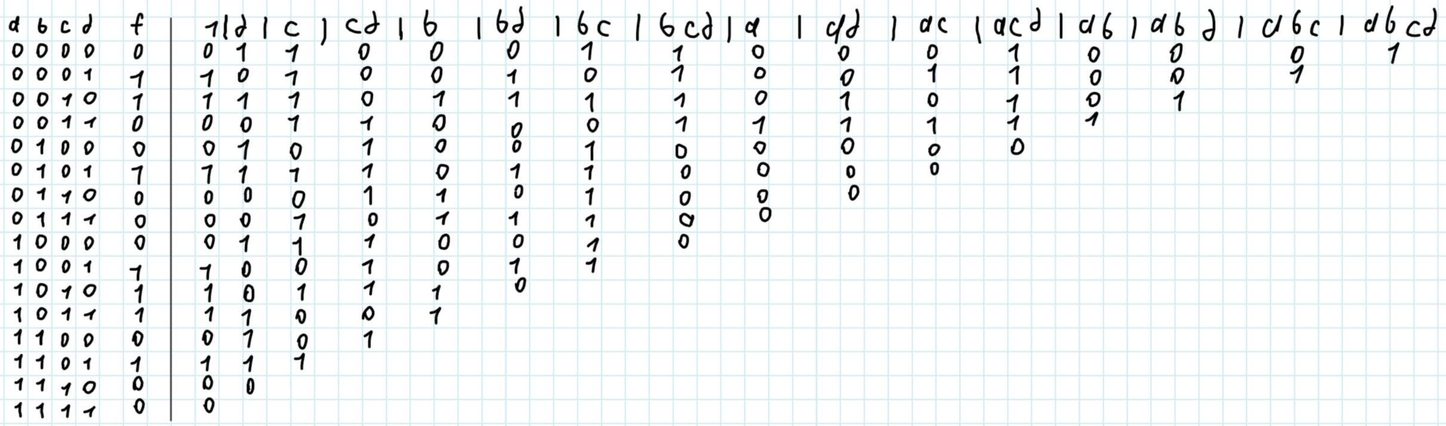


mDNF = $\bar{c}d \vee \bar{b}cd \vee a\bar{b}c$

mCNF = $(\bar{b} \vee \bar{c}) \wedge (c \vee d) \wedge (a \vee \bar{c} \vee \bar{d})$

BCF = $\bar{c}d \vee a\bar{b}d \vee a\bar{b}c \vee \bar{b}c\bar{d}$

a	b	c	d	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0



ANF = $d \oplus c \oplus bc \oplus bcd \oplus acd \oplus abcd$

3. Convert the following formulae to CNF.

(a) $X \leftrightarrow (A \wedge B)$

(b) $Z \leftrightarrow \bigvee_i C_i$

(c) $D_1 \oplus \cdots \oplus D_n$

(d) majority(X_1, X_2, X_3) [1]

(e) $R \rightarrow (S \rightarrow (T \rightarrow \bigwedge_i F_i))$

(f) $M \rightarrow (H \leftrightarrow \bigvee_i D_i)$

a)

A \ BX	00	01	11	10
0	1	0	0	1
1	1	0	1	0

$mCNF = (a \vee \bar{x}) \wedge (b \vee \bar{x}) \wedge (\bar{a} \vee \bar{b} \vee x)$

b) $Z \leftrightarrow \bigvee_i c_i$

$(\bar{z} \vee \bigvee_i c_i) \wedge (z \wedge \overline{\bigvee_i c_i})$

$(\bar{z} \vee \bigvee_i c_i) \wedge z \wedge \bigwedge_i \bar{c}_i$

c) $d_1 \oplus \ldots \oplus d_n$

$d_1 \oplus d_2$
$(d_1 \vee d_2) \wedge (\bar{d}_1 \vee \bar{d}_2)$

$((d_1 \vee d_2) \wedge (\bar{d}_1 \vee \bar{d}_2) \vee d_3) \wedge (\overline{(d_1 \vee d_2) \wedge (\bar{d}_1 \vee \bar{d}_2)} \vee \bar{d}_3)$

$(d_1 \vee d_2 \vee d_3) \wedge (d_1 \vee \bar{d}_2 \vee \bar{d}_3) \wedge (\bar{d}_1 \vee d_2 \vee \bar{d}_3) \wedge (\bar{d}_1 \vee \bar{d}_2 \vee d_3)$

Аналогично для 4-ех:

$(d_1 \vee d_2 \vee d_3 \vee d_4) \wedge (d_1 \vee d_2 \vee \bar{d}_3 \vee \bar{d}_4) \wedge (d_1 \vee \bar{d}_2 \vee d_3 \vee \bar{d}_4) \wedge (d_1 \vee \bar{d}_2 \vee \bar{d}_3 \vee d_4) \wedge$
$(\bar{d}_1 \vee d_2 \vee d_3 \vee \bar{d}_4) \wedge (\bar{d}_1 \vee d_2 \vee \bar{d}_3 \vee d_4) \wedge (\bar{d}_1 \vee \bar{d}_2 \vee d_3 \vee d_4) \wedge (\bar{d}_1 \vee \bar{d}_2 \vee \bar{d}_3 \vee \bar{d}_4)$

Для n переменных будет 2^{n-1} конъюнктов со всеми перестановками наборов, где чётное количество ¬

d) Вектор значений $f_{majority3}$ — 0 0 0 1 0 1 1 1

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$mCNF = (X_1 \vee X_2) \wedge (X_1 \vee X_3) \wedge (X_2 \vee X_3)$

e) $R \rightarrow (S \rightarrow (T \rightarrow \bigwedge_i F_i))$

$\bar{R} \vee \bar{S} \vee \bar{T} \vee \bigwedge_i F_i$

$(F_1 \vee \bar{R} \vee \bar{S} \vee \bar{T}) \wedge (F_2 \vee \bar{R} \vee \bar{S} \vee \bar{T}) \ldots$

$\bigwedge_i (F_i \vee \bar{R} \vee \bar{S} \vee \bar{T})$

f) $M \rightarrow (H \leftrightarrow \bigvee_i D_i)$

$M \rightarrow ((\bar{H} \vee \bigvee_i D_i) \wedge H \wedge \bigwedge_i D_i)$

$\bar{M} \vee ((\bar{H} \vee \bigvee_i D_i) \wedge H \wedge \bigwedge_i D_i)$

$M \wedge ((\bar{H} \vee \bigvee_i D_i) \wedge H \wedge \bigwedge_i D_i)$

c) В качестве примера рассмотрим случай с 10 переменными:

!!! эту часть НЕ нужно проверять Я просто захотел отвлечься

Вместо таблицы истинности я, пожалуй, просто запишу вектор функции

v =

0110100110010110100101100110100110010110011010010110100110010110100101100110100101101001100101100110100110010110100101100110100110010110011010010110100
1100101100110100110010110100101100110100101101001100101101001011001101001100101101001011001101001011010011001011001101001
1010010110011010010110100110010110011010010110100110010110011010011001011001101001011010011001011010010110011010010110100110010110100101100110100101101001100101101001011001101001011010010110011010010
1100110100101101001100101100110100110010110100101101001100101100110100101101001100101101001011001101001100101100110100110010110100101101001100101100110100101101001011001101001011001100110
1001100101100110100101101001100101100110100110010110100101101001011001101001100101100110100101101001100101101001011001101001011010011001011001101001100101101001011010010110011010010110011000

v.count sketch aside, this block is what the page prints as the function vector (its lines as wrapped on the page):

При помощи `v.count('0')`, нетрудно заметить, что у нас 512 наборов значений, где функция ложна. Тогда при построении СКНФ у нас будет 512 конъюнктов ($512 = 2^{10-1}$)

При помощи more magic python, нетрудно получить все конъюнкты и записать СКНФ

[illegible]

↑ все начальные значения при которых $f_{10} = 0$

Для каждого набора составим дизъюнкцию элементов, если элементу соответствует 1, то берём его отрицание. Конъюнкция всех этих наборов и будет СКНФ (inkodo начинает очень лагать если я вставляю СКНФ, поэтому так)

Можно построить mCNF по карте Карно

И окажется, что mCNF совпадет с СКНФ

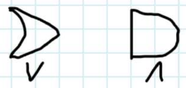
4. For each given system of functions F_i, determine whether it is functionally complete using Post's criterion. For each basis F_i, use it to represent the majority(A, B, C) function. Draw a combinational Boolean circuit for each resulting formula.
   (a) $F_1 = \{\wedge, \vee, \neg\}$
   (b) $F_2 = \{f_{14}^{(2)}\}$
   (c) $F_3 = \{\rightarrow, \not\rightarrow\}$
   (d) $F_4 = \{1, \leftrightarrow, \wedge\}$

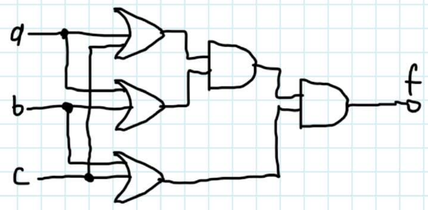
a)

	T_0	T_1	L	M	S
$\wedge$	+	+	−	+	−
$\vee$	+	+	−	+	−
$\neg$	−	−	+	−	+

F_1 is complete

majority$(a,b,c) = (a \vee b) \wedge (a \vee c) \wedge (b \vee c)$





b)

	T_0	T_1	L	M	S
$\uparrow$	−	−	−	−	−

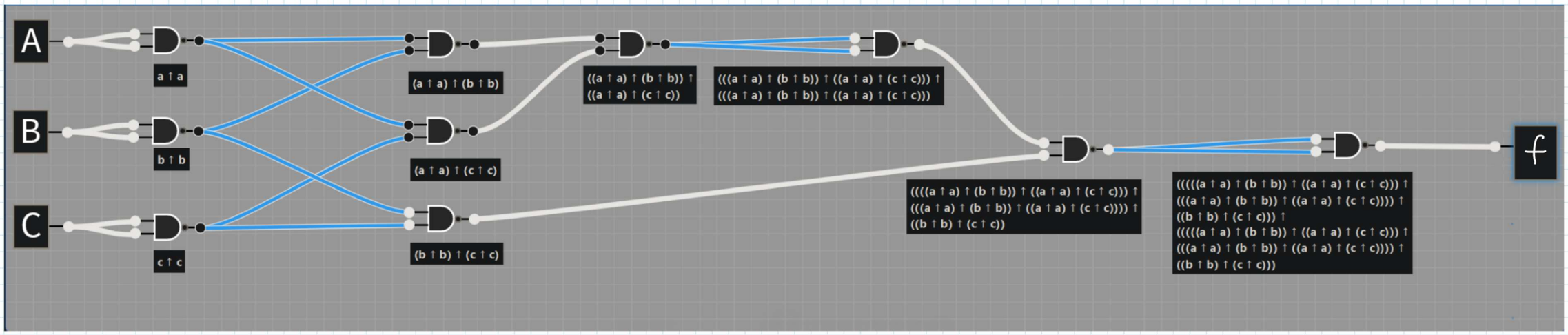
F_2 is complete

$f_{14}^{(2)} = \bar{a} \vee \bar{b} = \overline{a \wedge b} = NAND = \uparrow$

$(a \uparrow a) \uparrow (b \uparrow b) = a \vee b$ $\quad$ $(a \uparrow b) \uparrow (a \uparrow b) = a \wedge b$

majority(a,b,c) = (((((a ↑ a) ↑ (b ↑ b)) ↑ ((a ↑ a) ↑ (c ↑ c))) ↑ (((a ↑ a) ↑ (b ↑ b)) ↑ ((a ↑ a) ↑ (c ↑ c)))) ↑ ((b ↑ b) ↑ (c ↑ c))) ↑ (((((a ↑ a) ↑ (b ↑ b)) ↑ ((a ↑ a) ↑ (c ↑ c))) ↑ (((a ↑ a) ↑ (b ↑ b)) ↑ ((a ↑ a) ↑ (c ↑ c)))) ↑ ((b ↑ b) ↑ (c ↑ c)))

$(a \vee b) \wedge (a \vee c)$ $\quad$ $b \vee c$



c)

	T_0	T_1	L	M	S
$\rightarrow$	−	+	−	−	−
$\not\rightarrow$	+	−	−	−	−

F_3 is complete

$a \rightarrow (a \not\rightarrow b) = a \uparrow b$ (можно представить функцию как в 2с)

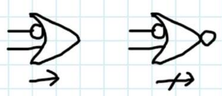
$b \rightarrow a \rightarrow a = a \vee b$

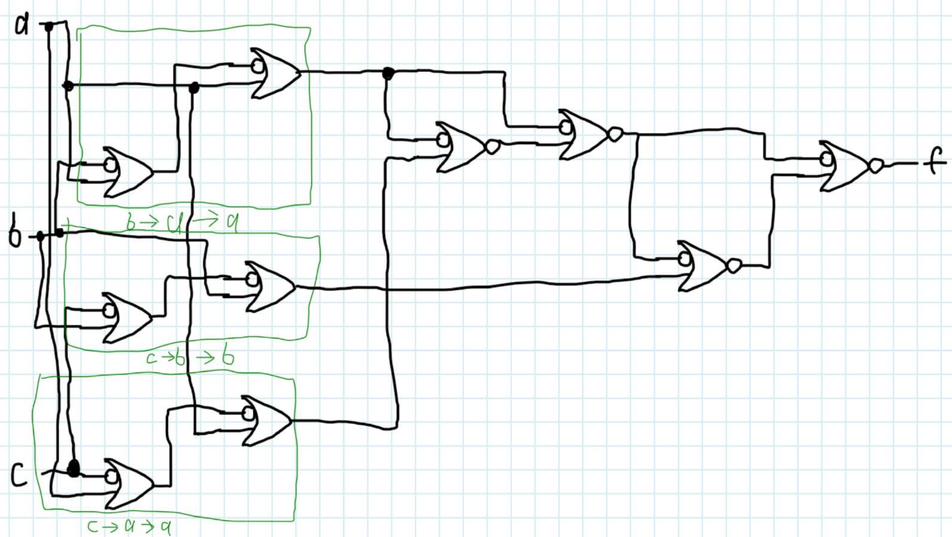
$a \not\rightarrow (a \not\rightarrow b) = a \wedge b$

$(b \rightarrow a \rightarrow a) \not\rightarrow ((b \rightarrow a \rightarrow a) \not\rightarrow (c \rightarrow a \rightarrow a))$

$(a \vee b) \wedge (a \vee c)$

majority$(a,b,c) = ((b \rightarrow a \rightarrow a) \not\rightarrow ((b \rightarrow a \rightarrow a) \not\rightarrow (c \rightarrow a \rightarrow a))) \not\rightarrow (((b \rightarrow a \rightarrow a) \not\rightarrow ((b \rightarrow a \rightarrow a) \not\rightarrow (c \rightarrow a \rightarrow a))) \not\rightarrow (c \rightarrow b \rightarrow b))$





4. For each given system of functions F_i, determine whether it is functionally complete using Post's criterion. For each basis F_i, use it to represent the majority (A, B, C) function. Draw a combinational Boolean circuit for each resulting formula.
   (a) $F_1 = \{\wedge, \vee, \neg\}$
   (b) $F_2 = \{f_{14}^{(2)}\}$
   (c) $F_3 = \{\rightarrow, \not\rightarrow\}$
   (d) $F_4 = \{1, \leftrightarrow, \wedge\}$

d)

	T_0	T_1	L	M	S
1	−	+	+	+	−
$\leftrightarrow$	−	+	+	−	−
$\wedge$	+	+	−	+	−

F_4 is not complete

5. Show — without using Post's criterion — that the Zhegalkin basis $\{\oplus, \wedge, 1\}$ is functionally complete.

Мы знаем, что $\{\neg, \wedge, \vee\}$ – functionally complete basis.

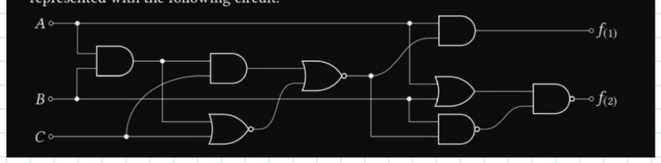
Выразим его с помощью базиса Жегалкина

$\bar{a} = 1 \oplus a$

$a \vee b = 1 \oplus ((1 \oplus a) \wedge (1 \oplus a))$ $\quad (a \vee b = \overline{\bar{a} \wedge \bar{b}})$

Таким образом базис Жегалкина – functionally complete basis.

6. Compute the truth table for the function $f\colon \mathbb{B}^3 \to \mathbb{B}^2$ (with the semantics $\langle A, B, C\rangle \mapsto \langle f_{(1)}, f_{(2)}\rangle$) represented with the following circuit.



$f_1 = a \wedge ((((a \wedge b \wedge c) \downarrow (a \wedge b)) \downarrow c))$

$f_2 = (b \uparrow (a \wedge (((a \wedge b \wedge c) \downarrow (a \wedge b)) \downarrow c)))) \uparrow (a \vee b)$

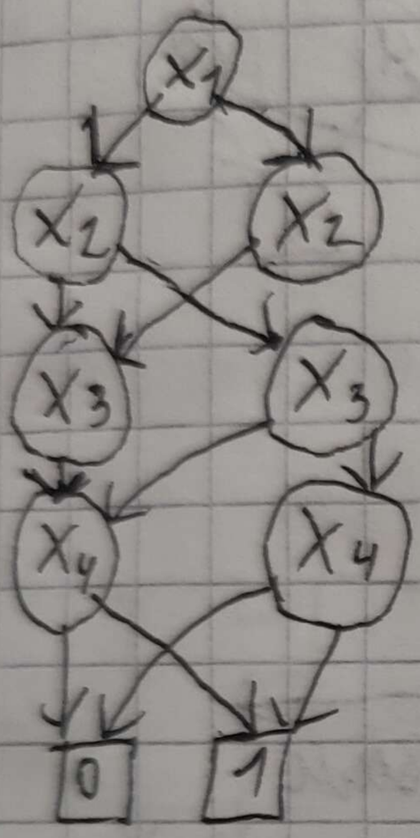
a	b	c	f_1	f_1
0	0	0	0	1
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

11. Consider a Boolean function ITE: $\mathbb{B}^3 \to \mathbb{B}$ defined as follows: $\mathrm{ITE}(c, x, y) = \begin{cases} x & \text{if } c=0 \\ y & \text{if } c=1 \end{cases}$. Construct a formula for it using the standard Boolean basis $\{\wedge, \vee, \neg\}$. Determine whether the set $\{\mathrm{ITE}\}$ is functionally complete.

№ 12

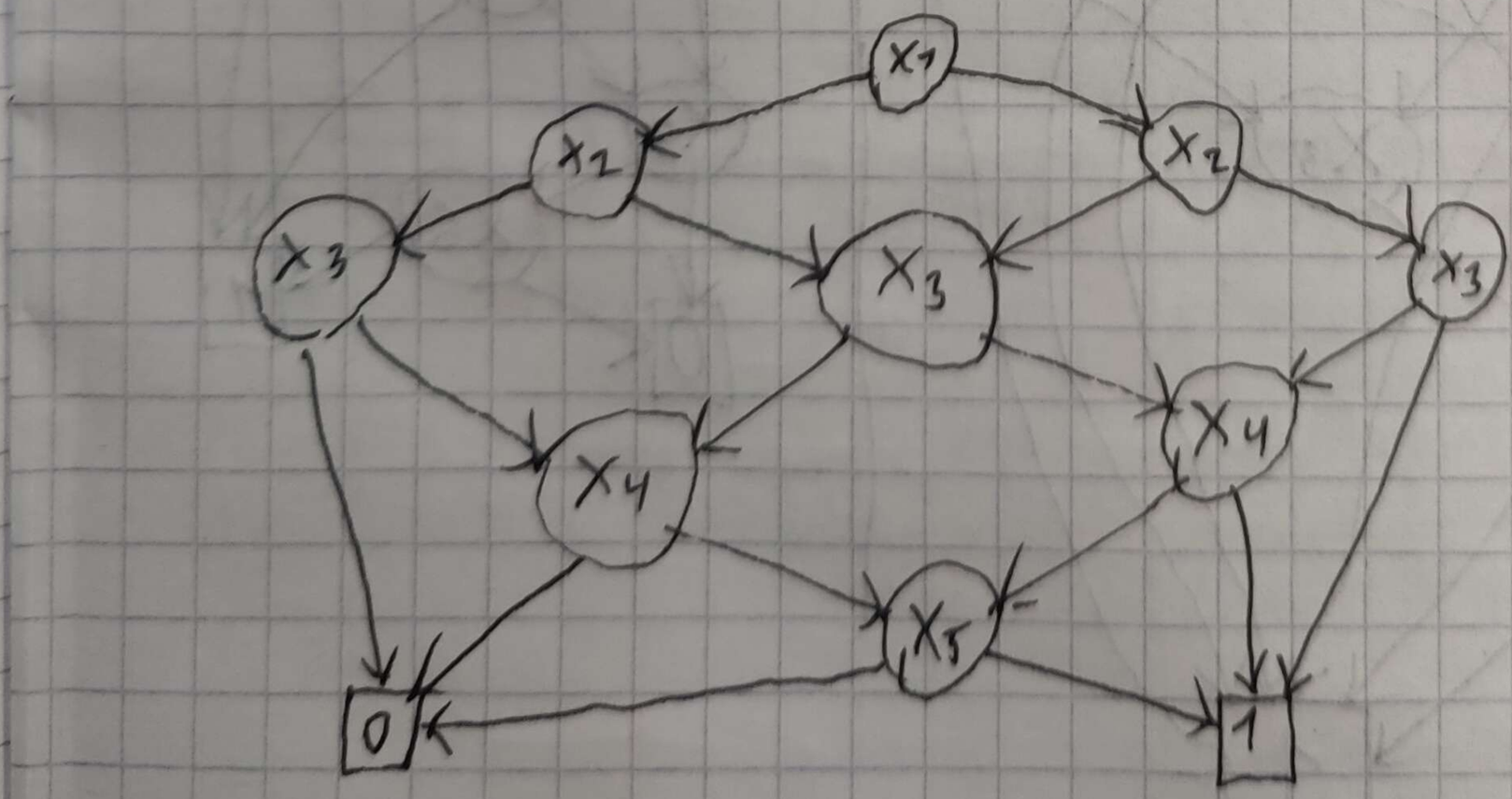
а) $f_1 = x_1 \oplus x_2 \oplus x_3 \oplus x_4$

ROBDD нельзя уменьшить изменив порядок переменных, т.к. xor – коммутативен

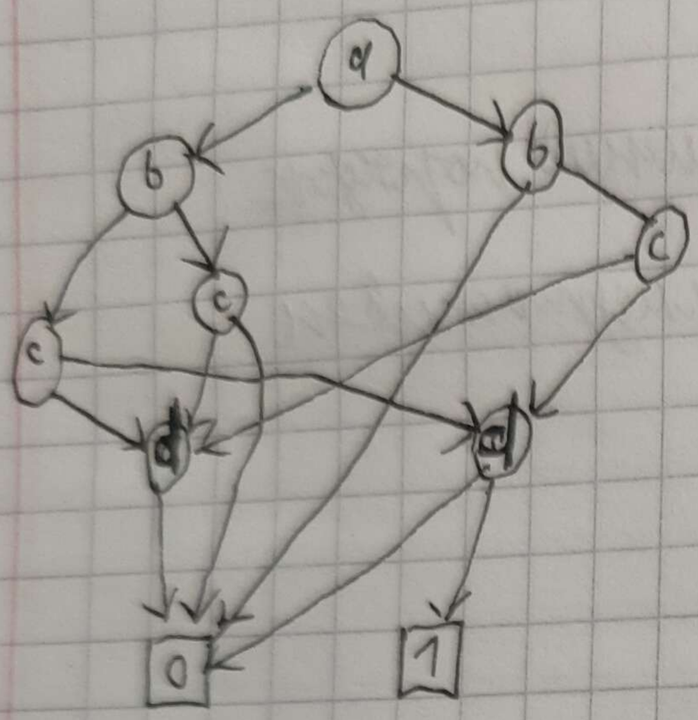


б) $f_2 = majority(x_1, x_2, x_3, x_4, x_5)$

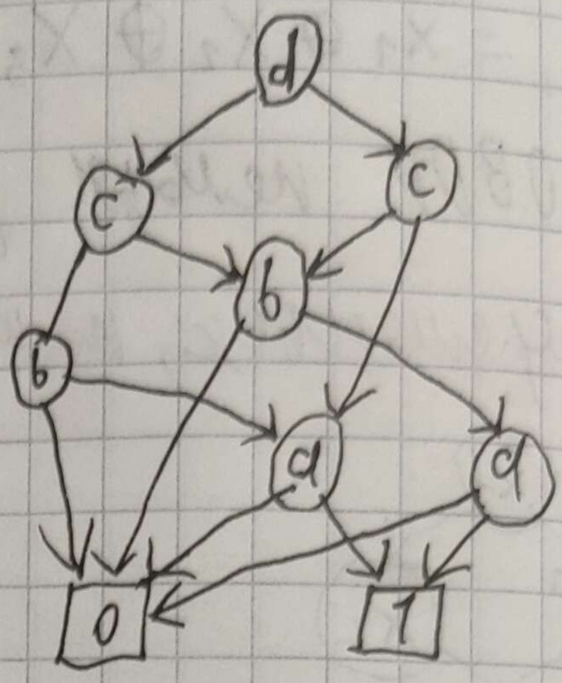
ROBDD нельзя уменьшить, т.к. majority не зависит от порядка аргументов



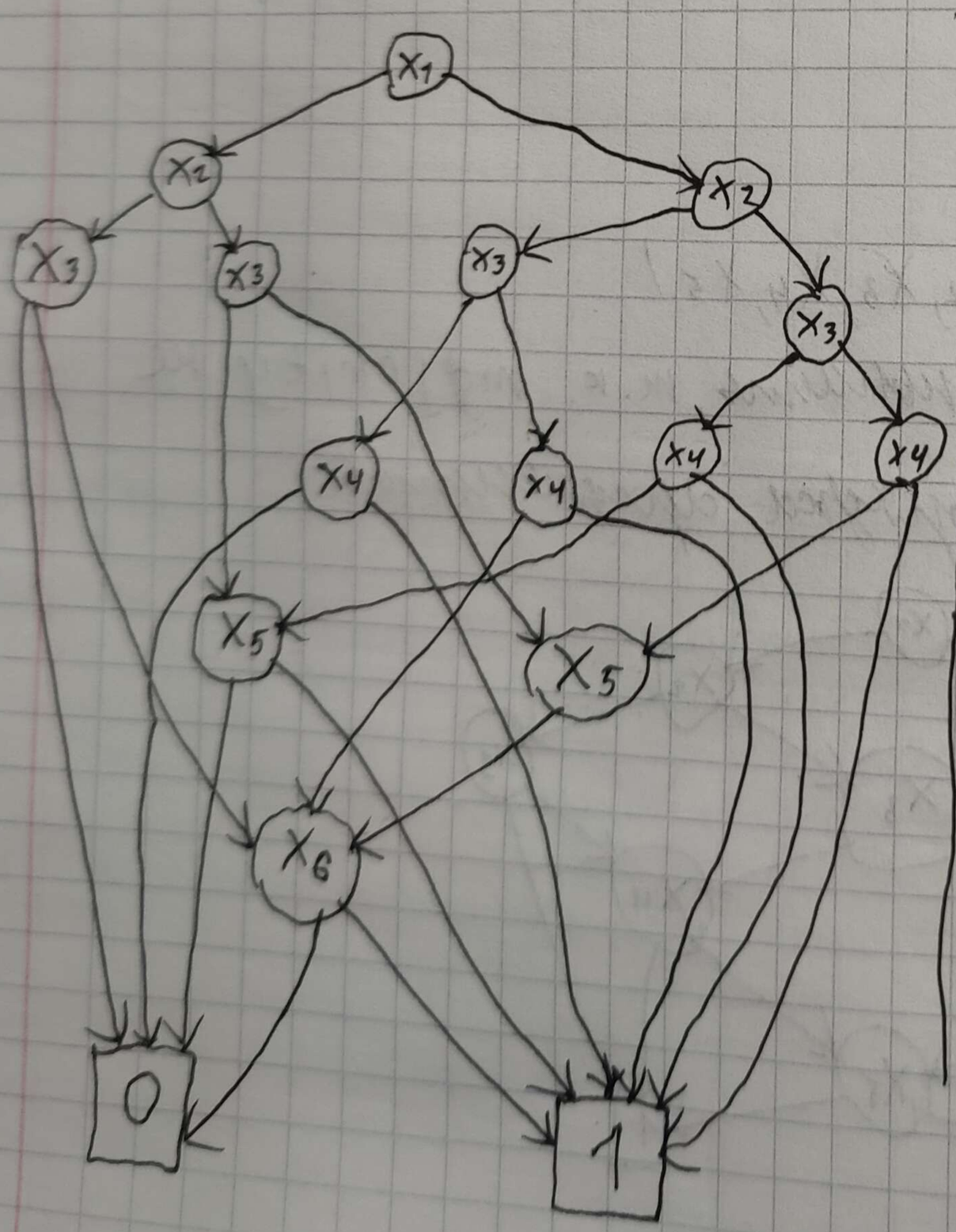
c) $f_3 = m(1,2,5,12,15) = \bar{a}\bar{b}\bar{c}d \vee \bar{a}\bar{b}c\bar{d} \vee \bar{a}b\bar{c}d \vee ab\bar{c}\bar{d} \vee abcd$



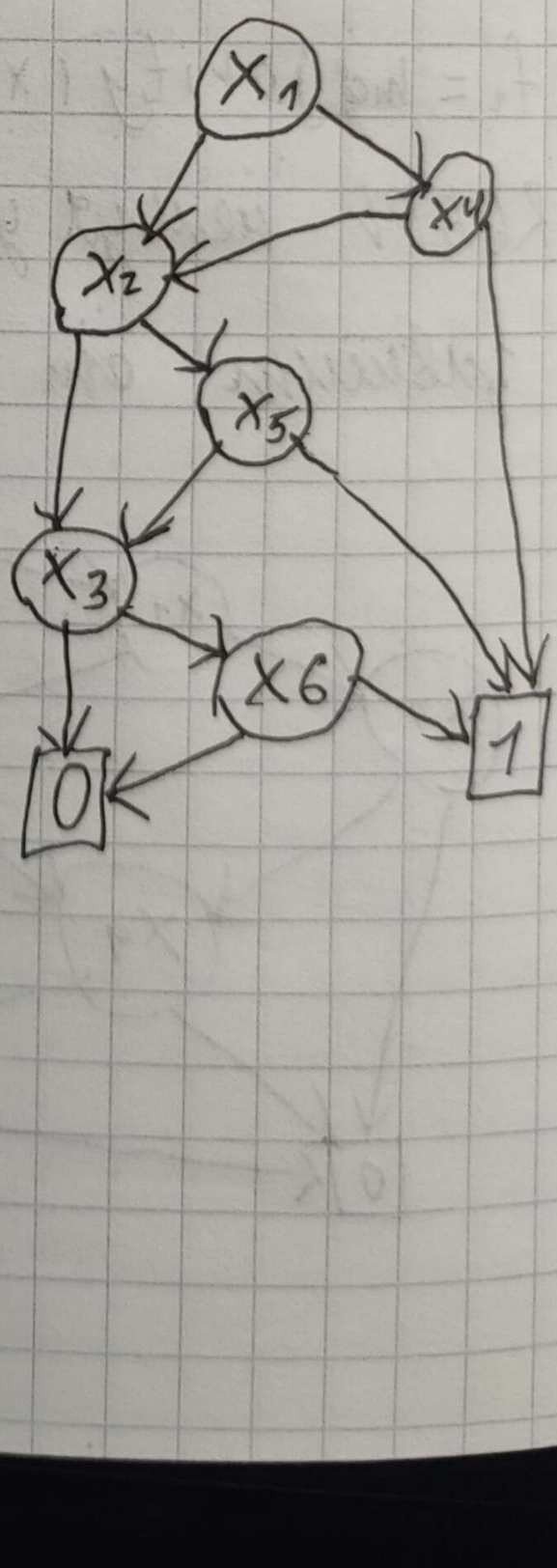
переставим
dcba



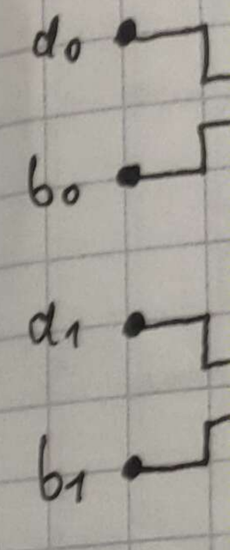
д) $f_4 = x_1x_4 \vee x_2x_5 \vee x_3x_6$



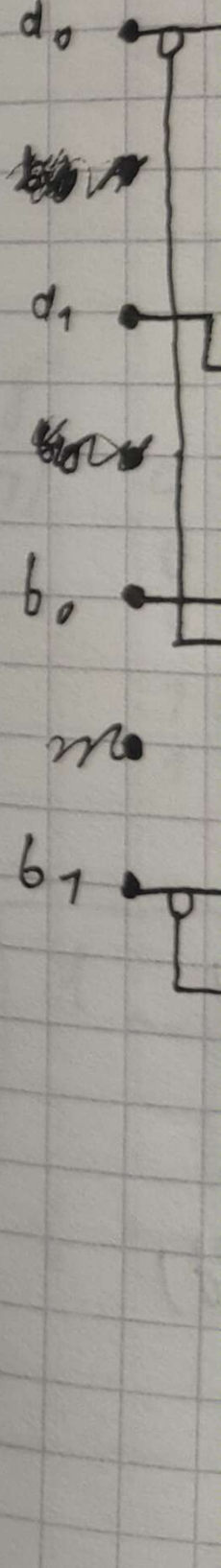
переставим
$x_1x_4x_2x_5x_3x_6$



9.

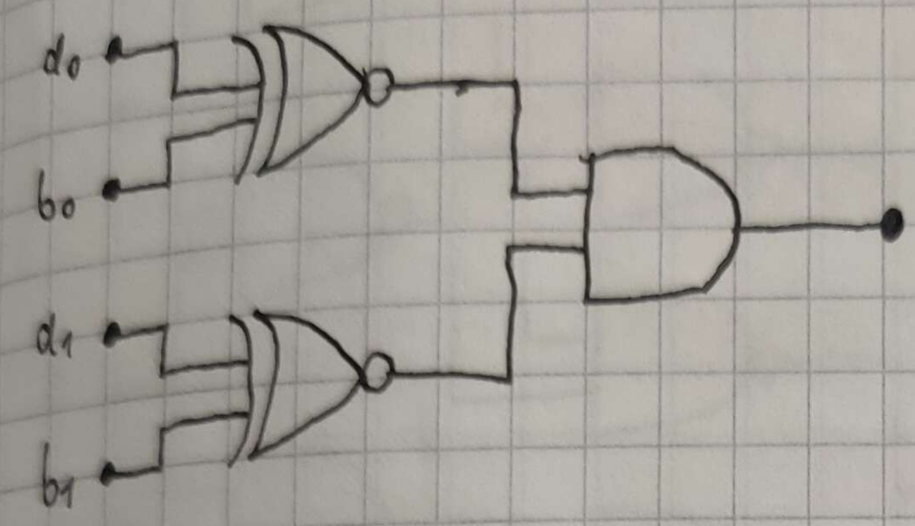


10.

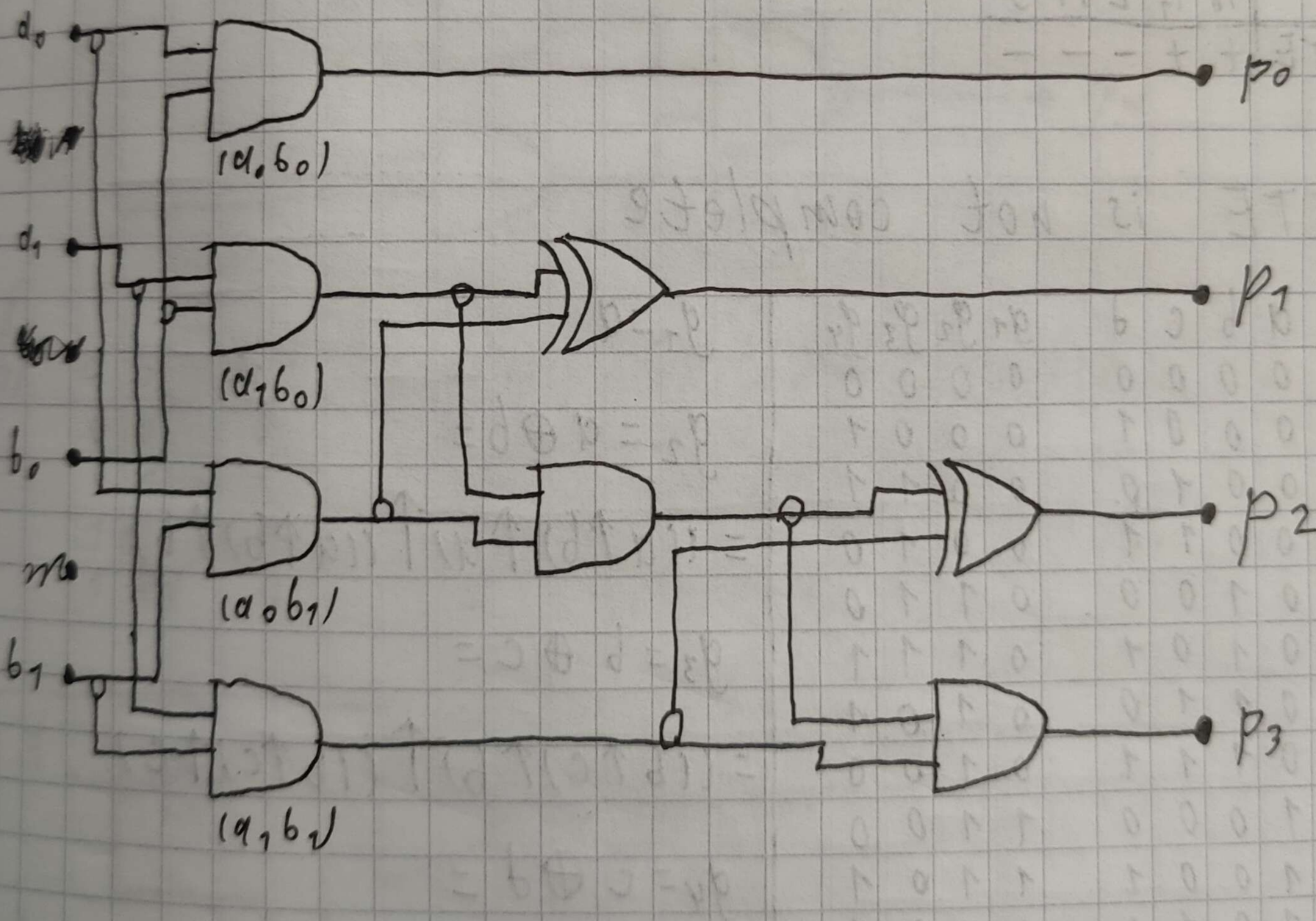


$\bar{c}\bar{d}va\bar{b}cd$

9. compare 2-bit integers



10. multiply 2-bit nums



11. $ITE(c,x,y) = \begin{cases} x, & \text{if } c=0 \\ y, & \text{if } c=1 \end{cases}$

c	x	y	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$ITE(c,x,y) = cy \vee \bar{c}x$

	T_0	T_1	L	M	S
ITE	+	+	−	−	−

ITE is not complete

7.

a	b	c	d	g_1	g_2	g_3	g_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

$g_1 = a$

$g_2 = a \oplus b =$

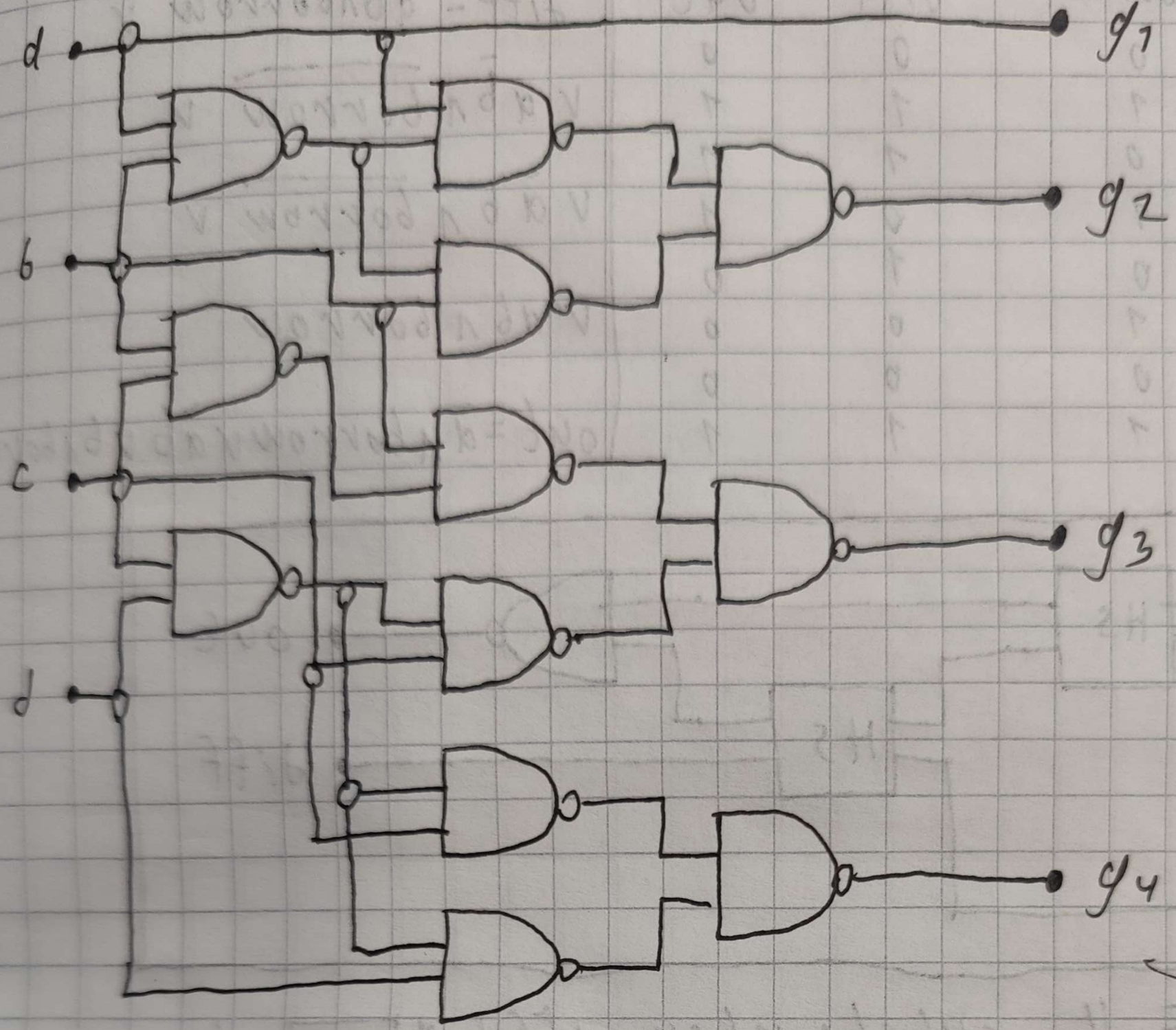
$= ((a \uparrow b) \uparrow a) \uparrow ((a \uparrow b) \uparrow b)$

$g_3 = b \oplus c =$

$= ((b \uparrow c) \uparrow b) \uparrow ((b \uparrow c) \uparrow c)$

$g_4 = c \oplus d =$

$= ((c \uparrow d) \uparrow c) \uparrow ((c \uparrow d) \uparrow d)$



8.

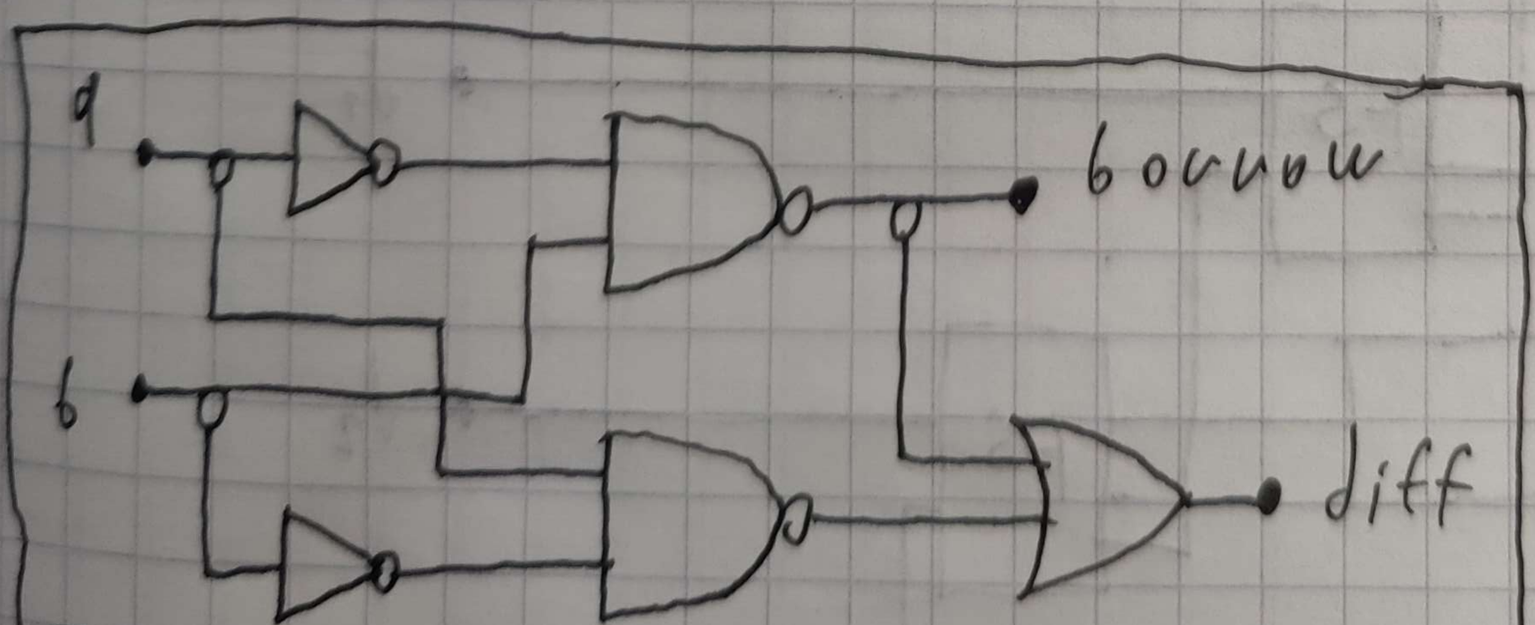
a)

a	b	diff	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

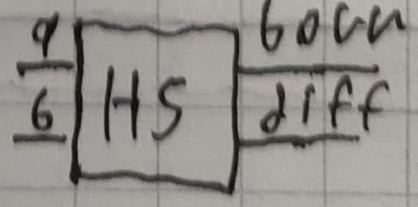
Half subtractor

$diff = \bar{a}b \vee \bar{b}a$

$borrow = \bar{a}b$



Half substraction (HS)

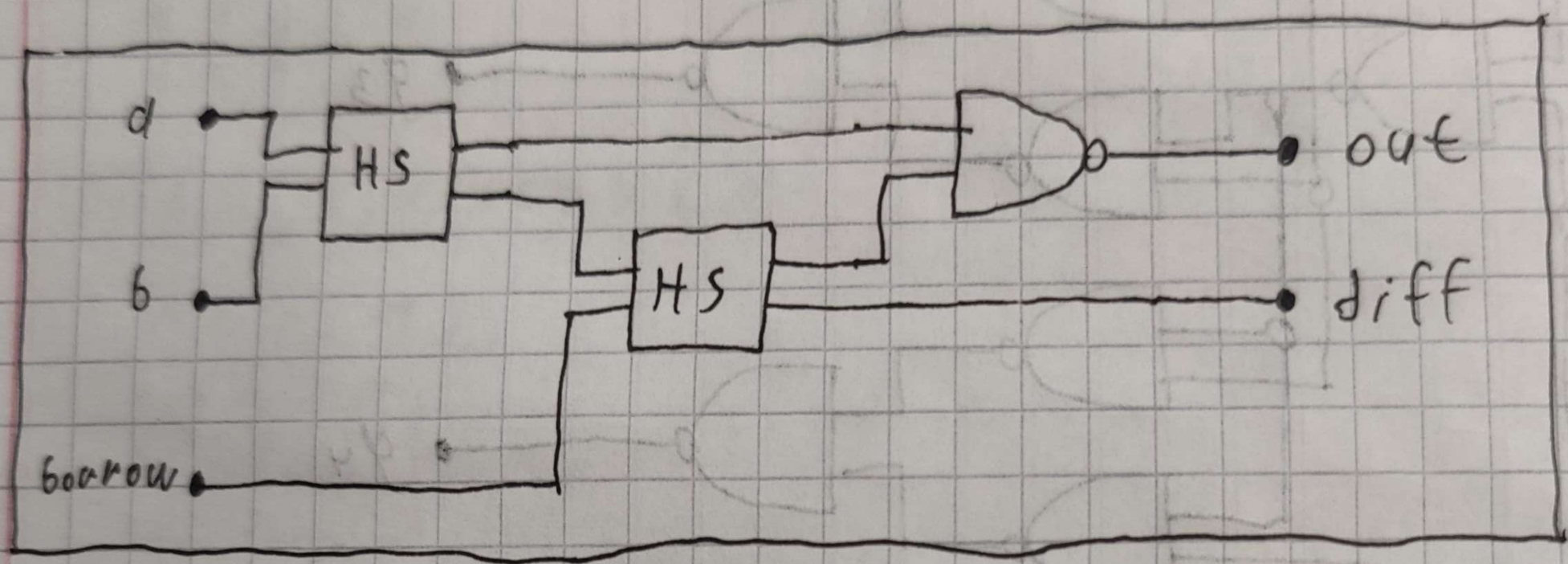


b)

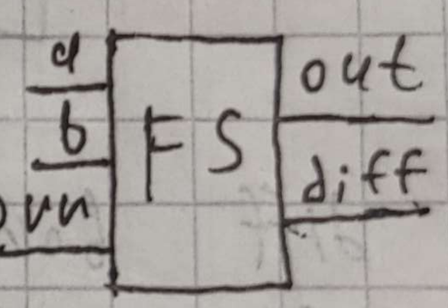
a	b	borrow	diff	out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$diff = \bar{a}\bar{b} \wedge borrow \vee a\bar{b} \wedge \overline{borrow} \vee \bar{a}b \wedge \overline{borrow} \vee ab \wedge borrow$

$out = \bar{a} \wedge borrow \vee ab \vee b \wedge borrow$



Full substractor (FS)



c)