Дополнения. Ликбез по информатике-2017.

Д.1. Алгебра логики.

Д.2. Карты Карно.

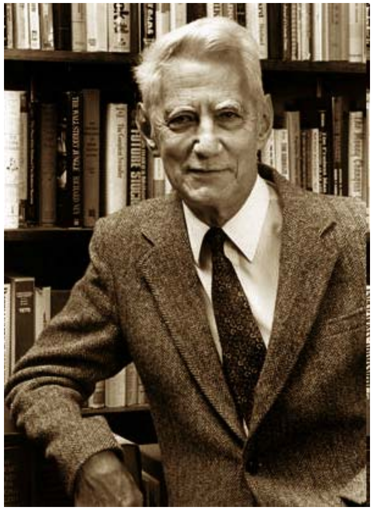
- **Алгебра** - раздел математики, изучающий общие свойства операций над элементами множества произвольной природы. Т.обр., задать алгебру - означает задать некоторое множество **элементов** (в частности, это м.б. переменные или константы), сформулировать исходные **аксиомы** и определяемое на них некоторое множество **операций** (базис алгебры).

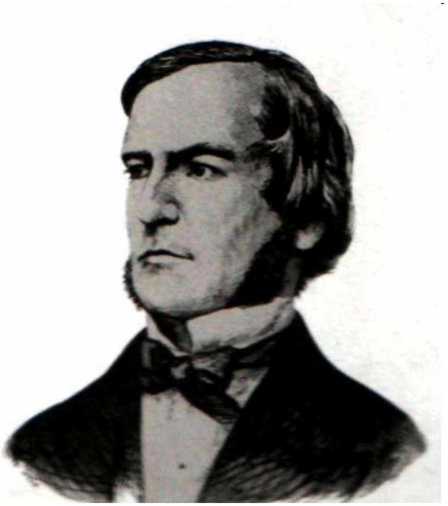
- Пример: алгебра, изучаемая в школе. Оперирует с множеством вещественных чисел. Базис – операции сложения, вычитания, умножения, деления, возведения в степень, извлечение корня. Множество операций в этой алгебре избыточно, т.к. их можно заменить другими, но это удобно.

- Особое место среди алгебр занимают двоичные алгебры: в частности, **булева алгебра** (Джордж Буль, математические методы логики, 1854 г., Клод Шеннон 1938 г., линейные сети переключателей).

- Булева алгебра и алгебра логики:

- *множества (элементы) 0 и 1;*
- *операции (базис алгебры);*
- *аксиомы.*





Булева алгебра :

Двоичные переменные: только два значения – 0 и 1.

Двоичные константы: только два значения – 0 и 1.

Двоичные функции: функции двоичных переменных тоже могут принимать только два значения – 0 и 1.

Область значений булевой функции: 0 и 1 (всего 2 значения)

Двоичные наборы: различные сочетания значений двоичных аргументов, число наборов определяется количеством аргументов

n=1 2 набора: 0 и 1

n=2 4 набора: 00, 01, 10 и 11

n=3 8 наборов: 000, 001, …, 111

Область определения булевой функции определяется разрядностью наборов множества элементов **из n аргументов: 2^n наборов**

Базис булевой алгебры:

- логическое сложение (дизъюнкция)
- логическое умножение (конъюнкция)
- отрицание (инверсия)

Таблица истинности: табличная форма задания двоичных функций.

В таблице истинности ф-ция задается своими значениями на всех наборах значений своих аргументов.
Т.е. это таблица, содержащая все возможные комбинации входных логических переменных и соответствующие им значения логических функций. Для n аргументов число сочетаний равно $N=2^n$, т.к. для каждого сочетания возможно 2 значения, то общее число функций составляет $2^N = 2^{2^n}$, в частности, для n=2 может быть 16 функций аргументов:

X2 X1	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
0 0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0 1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Названия логических функций 2-х переменных

Функция	Описание
F0	Константа нуль
F1	Отрицание дизъюнкции (ИЛИ-НЕ), функция Пирса
F2	Отрицание обратной импликации
F3	Отрицание X2
F4	Отрицание прямой импликации
F5	Отрицание X1
F6	Сложение по модулю 2
F7	Отрицание конъюнкции (И-НЕ), функция Ше́ффера
F8	Конъюнкции (И)
F9	Эквивалентность
F10	Повторение X1
F11	Обратная импликация
F12	Повторение X2
F13	Прямая импликация
F14	Дизъюнкция(ИЛИ)
F15	Константа единица

При помощи набора булевых функций 2-х аргументов можно описать **любую цифровую систему**.

На практике используют не все 16 ф-ций, а только некоторые, которые обеспечивают представление любой другой функции.

Набор таких ф-ций называется функционально полным набором (ФПН).

Основной ФПН:

ИЛИ, И, НЕ

другие ФПН:

ИЛИ-НЕ (ф-ция Пирса)

И-НЕ (ф-ция Шеффера)

Синтез логических устройств в базисе ФПН состоит из представления этих функций в **нормальных формах** и **минимизации.**

Пять аксиом алгебры логики :

A1:	$X = 0$, если $X \neq 1$	($\bar{A}$1):	$X = 1$, если $X \neq 0$
A2:	если $X = 0$, то $\sim X = 1$	($\bar{A}$2):	если $X = 1$, то $\sim X = 0$
A3:	$0 * 0 = 0$	($\bar{A}$3):	$1 + 1 = 1$
A4:	$1 * 1 = 1$	($\bar{A}$4):	$0 + 0 = 0$
A5:	$0 * 1 = 0$ и $1 * 0 = 0$	($\bar{A}$5):	$1 + 0 = 1$ и $0 + 1 = 1$

Другие законы (теоремы) алгебры логики могут быть выведены их этих аксиом

Метод полной индукции: любую теорему можно доказать, убедившись в ее справедливости для всех возможных значений входящих в нее переменных

(т.е. построив таблицу истинности)

Основные законы булевой алгебры

$\sim 0 = 1$

$A + 0 = A$

$A + 1 = 1$

$A + A = A$

$A + A = 1$

$\sim 1 = 0$

$A \bullet 0 = 0$

$A \bullet 1 = A$

$A \bullet A = A$

$A \bullet A = 0$

Основные законы булевой алгебры

$$A + B = B + A \qquad A \cdot B = B \cdot A$$

$$A + A \cdot B = A \qquad (A + B) \cdot B = B$$

$$A + B \cdot C = (A + B) \cdot (A + C)$$

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

Основные законы булевой алгебры

Правило отрицания:

Для получения отрицания некоторого выражения достаточно заменить в нем:

1) знаки дизъюнкции знаками конъюнкции,
2) знаки конъюнкции знаками дизъюнкции,
3) все аргументы – их отрицаниями,
4) константы 0 и 1 на их противоположное значение

$$\boxed{\sim F(x,y,\dots,z)_{\&,V,0,1} = F(\sim x,\sim y,\dots,\sim z)_{V,\&,1,0}}$$

Примеры:

$$\sim(x_1 * x_2 * \dots * x_n) = (\sim x_1)+(\sim x_2)+\dots+(\sim x_n)$$

$$\sim(x_1 + x_2 + \dots + x_n) = (\sim x_1)*(\sim x_2)*\dots*(\sim x_n)$$

Булево выражение в виде **суммы произведений** наз. **дизъюнктивной нормальной формой (ДНФ)**

Булево выражение в виде **произведения сумм** наз. **конъюктивной нормальной формой (КНФ)**

Минтерм (конституэнта единицы): полное произведение всех входных переменных, соответствующее одной строке таблицы истинности, в которой значения функции равно логической 1.

Переменная входит в минтерм с инверсией, если ее значение в данной строке равно 0 и без инверсии, если ее значение в данной строке равно 1.

Совершенная дизъюнктивная нормальная форма (СДНФ) – каноническая сумма минтермов.

Число слагаемых в **СДНФ** равно числу строк таблицы истинности, в которых **функция принимает единичное значение.**

Каждое слагаемое есть полное произведение всех переменных (минтерм). Т.е. переменная входит в произведение без инверсии, если она в этой строке равна 1 и инвертируется, если ее значение в этой строке равно 0.

Справедливы теоремы:

Любую функцию двоичных переменных можно представить в виде суперпозиции (комбинации) конечного числа операций НЕ, И, ИЛИ над двоичными переменными из ее области определения.

Теорема построения СДНФ :

Для произвольной двоичной функции n переменнных F(x1,x2,…,xn), не равной тождественно нулю, справедливо следующее утверждение: если T1,T2,…,Tk – все точки ее определения, в которых F=1, то

$F(x1,x2,\ldots,xn)=f_1+f_2+\ldots+f_k$,

где $f_j=y_{j1}*y_{j2}*\ldots*y_{jn}$, причем: $y_{ji}=$ {x_i, если $x_i = 1$ в точке T_j и

{$\sim x_i$, если $x_i = 0$ в точке T_j

Теорема построения СКНФ:

Для произвольной двоичной функции n переменнных F(x1,x2,…,xn), не равной тождественно нулю справедливо следующее утверждение: если S1,S2,…,Sl – все точки ее определения, в которых F=0, то

$F(x1,x2,\ldots,xn)=g_1*g_2*\ldots*g_l$,

где $g_j=y_{i1}+y_{i2}+\ldots+y_{jm}$, причем: $y_{ji}=$ {x_i, если $x_i = 0$ в точке Sj и

{$\sim x_i$, если $x_i = 1$ в точке Sj

Алгоритм действий для построения СДНФ (дизъюнкция коньюнкций):

• составить таблицу истинности;

• для строк, в которых F=1 выписать произведения всех элементов строки взяв их отрицание, если элемент равен 0;

• СДНФ есть сумма полученных таким образом произведений

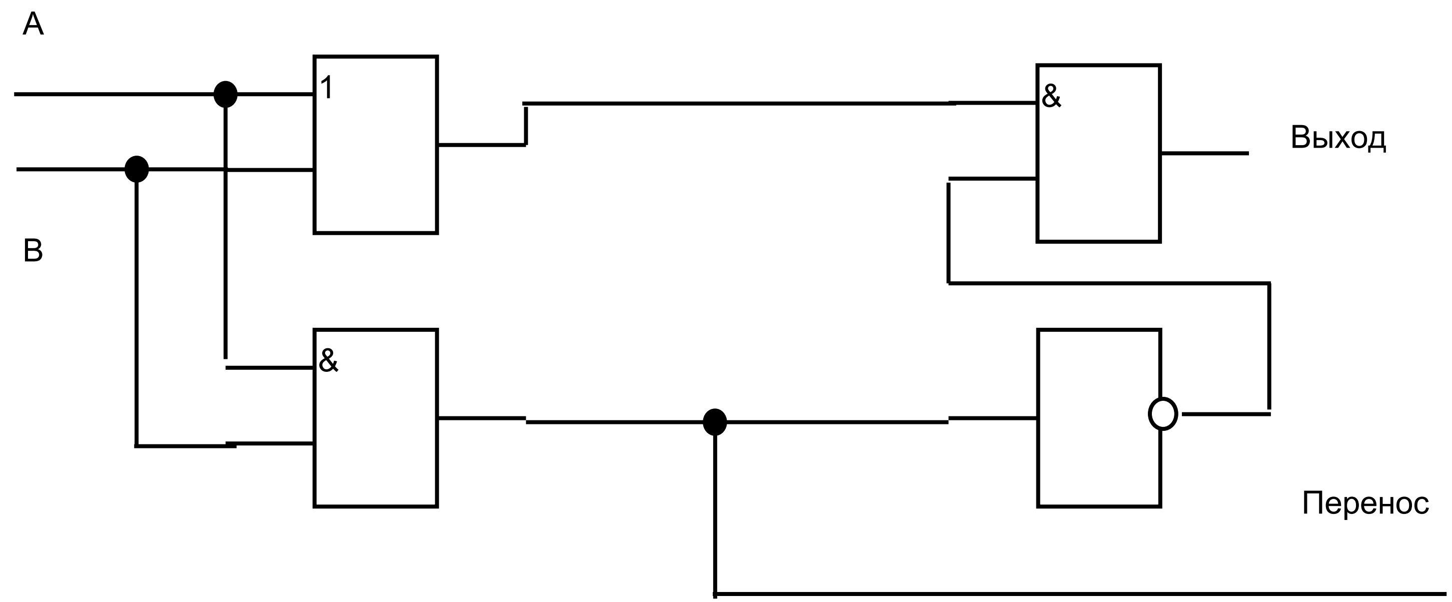
• упростить полученную логическую формулу, используя соотношения алгебры логики. Пример:

A	B	F	T	S
0	0	1	(~A)*(~B)	-
0	1	1	(~A)*B	-
1	0	0	-	(~A)+B
1	1	1	A*B	-

F=(~A)*(~B)+(~A)*B+A*B=(~A)*((~B)+B)+(A*B)=(~A)*1+A*B=((~A)+A))*((~A)+B)=
=(~A)+B

Т.к. (~A)+A=1 и A+BC=(A+B)(A+C)

!!! *Д.з.:* исходя из таблицы истинности, построить логические функции F и P, упростить выражения и доказать, что схема реализует это устройство



A	B	F	P
0 В («0»)	0 В («0»)	0 В («0»)	0 В («0»)
0 В («0»)	5 В («1»)	5 В («1»)	0 В («0»)
5 В («1»)	0 В («0»)	5 В («1»)	0 В («0»)
5 В («1»)	5 В («1»)	0 В («0»)	5 В («1»)

Простейший сумматор –пример схемы комбинационной логики

Yj = f(Xi)

Карты Карно: один из методов оптимизации логических функций, образованных из таблиц истинности

Графическое представление таблиц истинности
Каждой клетке соответствует одна строка таблицы истинности.
По осям карты расставляются сочетания переменных.

Смысл минимизации:

найти логические суммы прямого и инверсного значения переменных типа: A+(~A) = 1. Тогда эту сумму можно в выражении можно отбросить.

Пример: a*b*c+(~a)*b*c=(a+(~a))*b*c=b*c

Карты следует составлять так, чтобы по осям сочетания переменных образовывали **код Грея**, т.е. соседние сочетания переменных должны отличаться не более, чем на значение в одной позиции

Карты Карно: Правила разметки карты (таблица значений):

1) Начинать разметку можно с любого сочетания переменных
2) Вертикальная ось размечается независимо от горизонтальной
3) По вертикали или по горизонтали д.б. перечислены все сочетания переменных
4) Для соседних клеток карты сочетания переменных должны отличаться не более, чем одним разрядом (код Грея), причем крайние клетки тоже считаются соседними (образуется кольцо)

Карты Карно, правила минимизации:

1) В карту вписываются соответствующие сочетаниям переменных значения функций и далее рассматриваются только те клетки, где значения равны 1
2) Все клетки карты, где значения функции равны 1, обводятся прямоугольными контурами по следующим правилам:
3) Контуры должны быть прямоугольными и содержать количество единиц, кратное 2^n, т.е. 1, 2, 4, 8 и т.д.
4) Количества единиц в контуре д.б. максимальным. При этом контуры могут пересекаться между собой. Следует учитывать, что крайние строки и крайние столбцы являются соседними.
5) Количество контуров д.б. минимальным, но все единицы, т.ч. отдельно стоящие, д.б. охвачены контурами.
6) Для каждого контура следует записать минимальное выражение как логическую сумму логических произведений. При этом переменная входит в произведение с инверсией, если ее значение в данном контуре равно 0 и без инверсии, если ее значение равно 1.
7) В окончательном выражении для контура следует оставить только те, переменные, которые остаются внутри данного контура неизменными.
8) Минимизированная функция образуется как сумма выражений по контурам.

Карты Карно, пример минимизации:

Таблица истинности: СДНФ: F=(~a)b(~c)+(~a)bc+a(~b)(~c)+ab(~c)+abc => ???

№	a	b	c	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

a\bc	*10*	*00*	*01*	*11*
0	1 (2)	0 (0)	0 (1)	1 (3)
1	1 (6)	1 (4)	0 (5)	1 (7)

это один контур

Суммы минтермов по контурам:

I : (~a)b(~c)+(~a)bc+ab(~c)+abc => b (*не изменяется только* b)

II : ab(~c)+a(~b)(~c) => a(~c) (*не изменяются a и ~c*)

Минимальная сумма:

$\boxed{F=b+a(\sim c)}$

Карты Карно: 4-х разрядное устройство для обнаружения простых чисел, пример минимизации:

При заданной 4-х разрядной двоичной комбинации $N=N_3N_2N_1N_0$ на входе схема вырабатывает на выходе 1, если N=1,2,3,5,7,11,13 и
0 – при остальных сигналах на входе

№	N_3	N_2	N_1	N_0	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
0	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
…					
11	1	0	1	1	1
…					
13	1	1	0	1	1
…					

Каноническая сумма (СДНФ):

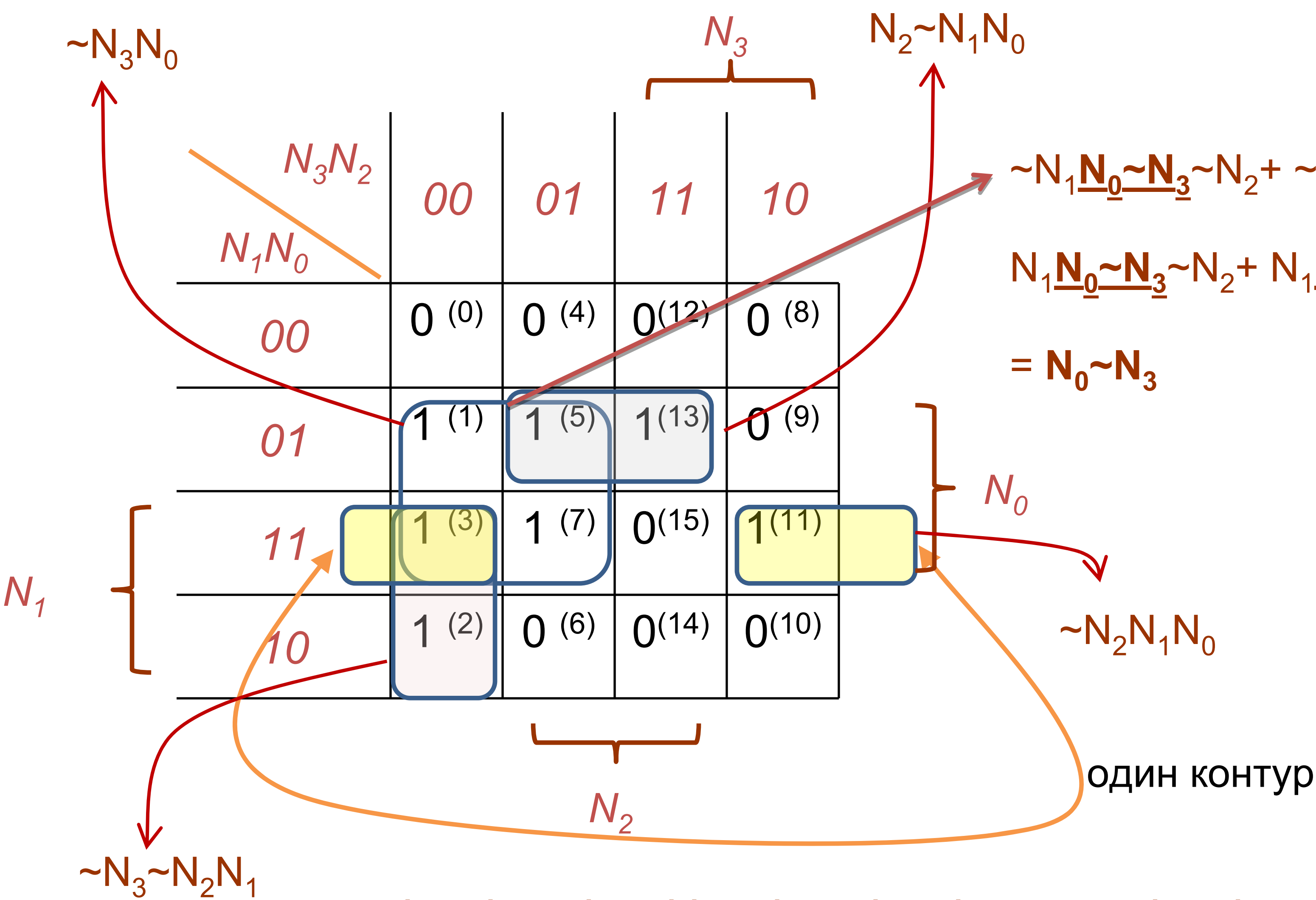
$$F=(\sim N_3)(\sim N_2)(\sim N_1)N_0+(\sim N_3)(\sim N_2)N_1(\sim N_0)+$$

$$+(\sim N_3)(\sim N_2)N_1N_0+(\sim N_3)N_2(\sim N_1)N_0+$$

$$+(\sim N_3)N_2N_1N_0+N_3(\sim N_2)N_1N_0+ N_3N_2(\sim N_1)N_0$$

Логическая схема:
4НЕ + 4 (4И) + 7 ИЛИ **(? ?)**

Карты Карно, 4-х разрядное устройство для обнаружения простых чисел, пример минимизации:



N_1N_0 \ N_3N_2	00	01	11	10
00	0 (0)	0 (4)	0 (12)	0 (8)
01	1 (1)	1 (5)	1 (13)	0 (9)
11	1 (3)	1 (7)	0 (15)	1 (11)
10	1 (2)	0 (6)	0 (14)	0 (10)

$\sim N_1\underline{\mathbf{N_0\sim N_3}}\sim N_2 + \sim N_1\underline{\mathbf{N_0\sim N_3}}N_2 +$

$N_1\underline{\mathbf{N_0\sim N_3}}\sim N_2 + N_1\underline{\mathbf{N_0\sim N_3}}N_2 =$

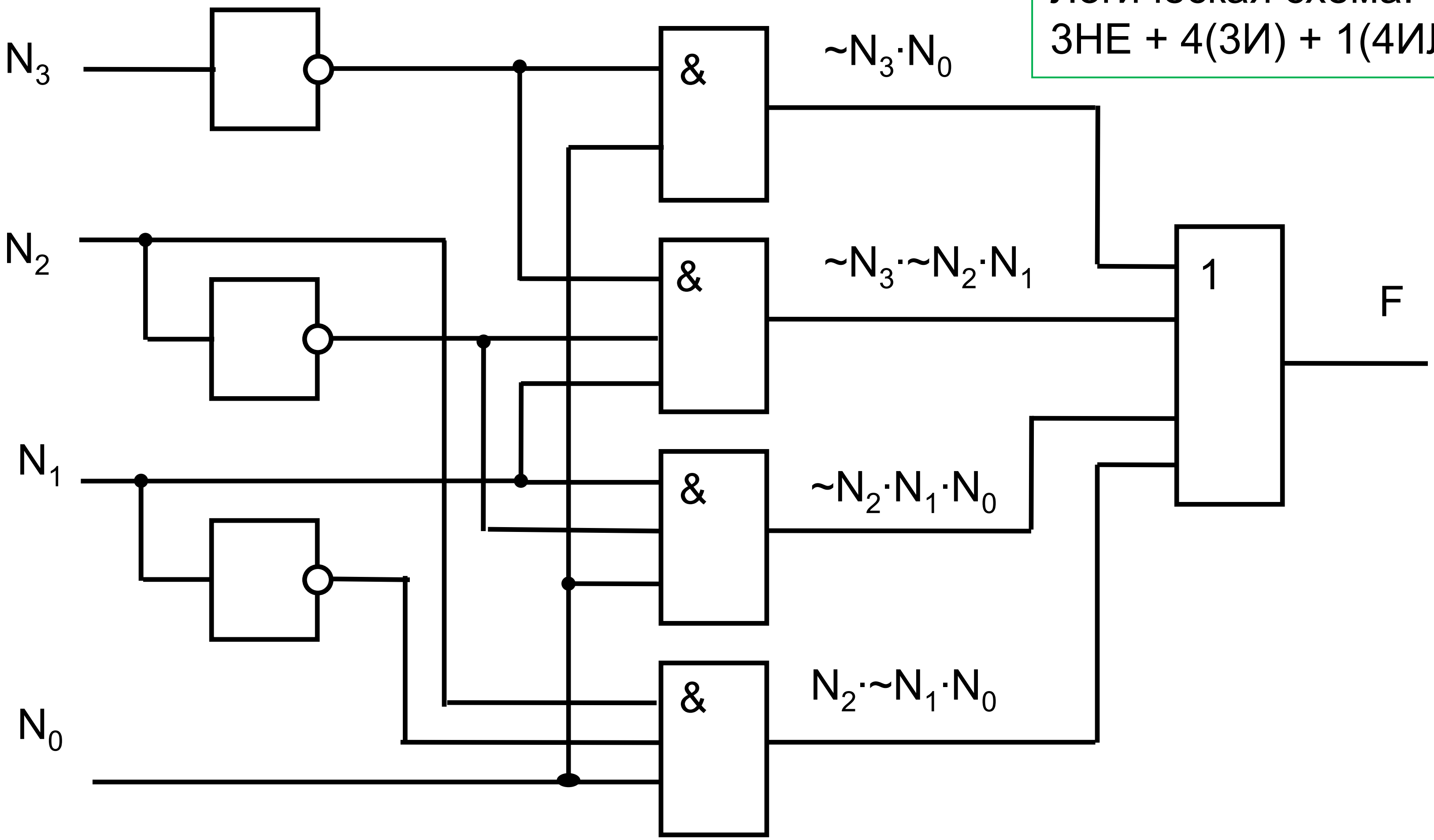
$= \mathbf{N_0\sim N_3}$

$$\mathbf{F=(\sim N_3)N_0+(\sim N_3)(\sim N_2)N_1+(\sim N_2)N_1N_0+N_2(\sim N_1)N_0}$$

Карты Карно, 4-х разрядное устройство для обнаружения простых чисел, пример минимизации:

$$F=(\sim N_3)N_0+(\sim N_3)(\sim N_2)N_1+(\sim N_2)N_1N_0+N_2(\sim N_1)N_0$$

Логическая схема:
3НЕ + 4(3И) + 1(4ИЛИ)



✍ ***Пример (д.з.):***
трехвходовая логическая схема, реализующая функцию
$x_{вых} = 3 * x_{вх}$

1) необходимая разрядность результата (максимально возможный код на выходе):

$3*(1\ 1\ 1) = 21_{10} = 10101_2$ => 5 разрядов (5 выходных функций)

2) Таблица истинности:

№	a	b	c	$X_в$	Q4	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	0	0	0
1	0	0	1	3	0	0	0	1	1
2	0	1	0	6	0	0	1	1	0
3	0	1	1	9	0	1	0	0	1
4	1	0	0	12	0	1	1	0	0
5	1	0	1	15	0	1	1	1	1
6	1	1	0	18	1	0	0	1	0
7	1	1	1	21	1	0	1	0	1

3) минимальное выражение для каждого выхода

Q4

a\bc	00	01	11	10
0	0 (0)	0 (1)	0 (3)	0 (2)
1	0 (4)	0 (5)	1 (7)	1 (6)

Q4 = abc+ab~c = ab

Q3

a\bc	00	01	11	10
0	0 (0)	0 (1)	1 (3)	0 (2)
1	1 (4)	1 (5)	0 (7)	0 (6)

Q3 = a~b~c+a~bc + ~abc
= a~b+~abc

Q2

a\bc	00	01	11	10
0	0 (0)	0 (1)	0 (3)	1 (2)
1	1 (4)	1 (5)	1 (7)	0 (6)

Q2 =
a~b~c+a~bc+a~bc+abc+~ab~c =
= a~b+ac+~ab~c

Q1

a\bc	00	01	11	10
0	0 (0)	1 (1)	0 (3)	1 (2)
1	0 (4)	1 (5)	0 (7)	1 (6)

Q1 =
~a~bc+a~bc+ ~ab~c+ ab~c =
= ~bc+b~c

Q0

1	00	01	11	10
0	0 (0)	1 (1)	1 (3)	0 (2)
1	0 (4)	1 (5)	1 (7)	0 (6)

Q0 =
~a~bc+a~bc+ ~abc+ abc =
= c

Схема (нарисовать дома):

Входы: a, b, c

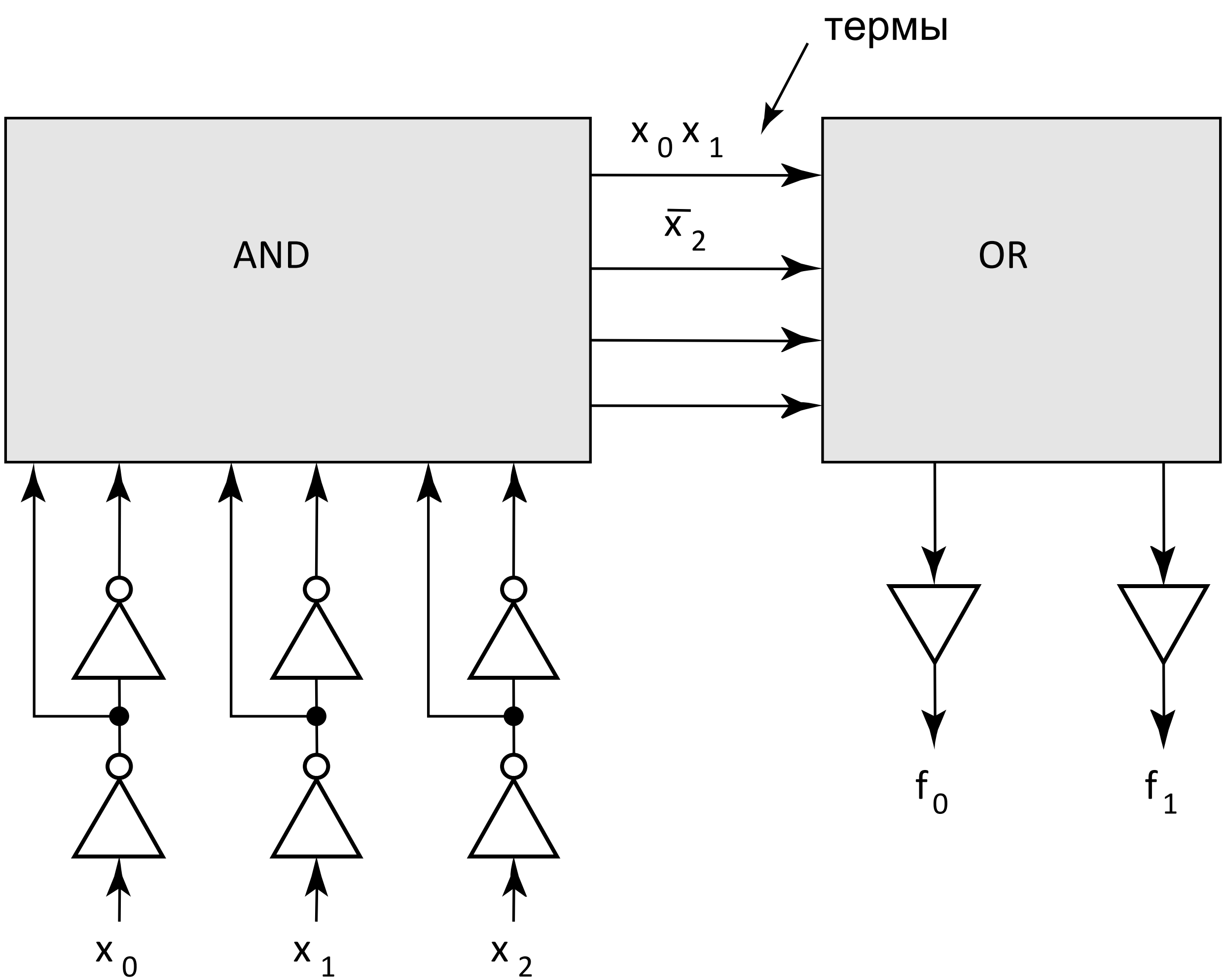
Требуются:

~a, ~b, ~c,

пары или тройки произведений (типа ab, ~ab, a~b и т.п.)

суммы произведений

✍ Обобщение:
Programmable Logic Array - PLA или ПЛМ



Что дальше? => ПЛИС = CPLD и FPGA

CPLD = complex programmable logic device

сложные программируемые логические устройства содержат относительно крупные программируемые логические блоки — макроячейки, соединённые с внешними выводами и внутренними шинами.

Функциональность CPLD кодируется в энергонезависимой памяти, поэтому нет необходимости их перепрограммировать при включении. Может применяться для расширения числа входов/выходов рядом с большими кристаллами, или для предобработки сигналов (например, контроллер COM-порта, USB, VGA).

FPGA = Field Programmable Gate Array содержат блоки умножения-суммирования, которые широко применяются при обработке сигналов (DSP, англ. *digital signal processing*), а также логические элементы (как правило, на базе таблиц перекодировки — таблиц истинности) и их блоки коммутации.Программа для FPGA хранится в распределённой памяти, которая может быть выполнена как на основе энергозависимых ячеек статического ОЗУ (подобные микросхемы производят, например, фирмы «Xilinx» и «Altera») — в этом случае программа не сохраняется при исчезновении электропитания микросхемы, так и на основе энергонезависимых ячеек flash-памяти или перемычек antifuse (такие микросхемы производит фирма «Actel» и «Lattice Semiconductor») — в этих случаях программа сохраняется при исчезновении электропитания. Если программа хранится в энергозависимой памяти, то при каждом включении питания микросхемы необходимо заново конфигурировать её при помощи начального загрузчика, который может быть встроен и в саму FPGA. FPGA применяются также, как ускорители универсальных процессоров в суперкомпьютерах (например, компьютер «Cray XD1» компании «Cray», проект «RASC» компании «Silicon Graphics» («SGI»)

Контрольные вопросы:
1.Что понимается под термином «проектирование» применительно к проектированию схем?
2.Что представляет собой современный проект ИС?
3.Назовите хотя бы 4 причины, обусловливающие необходимость автоматизации проектирования?
4.Перечислите наиболее крупные компании – производители САПР для СБИС.
5.Перечислите три основные методологии проектирования.
6.Что нужно определить, чтобы задать алгебру?
7.Элементы и операции булевой алгебры.
8.Аксиомы булевой алгебры.
9.Что такое таблица истинности?
10.Дайте определение минтерма.
11.Как по таблице истинности построить СДНФ?
12.Как по таблице истинности построить СКНФ?
13.Что такое функционально полной набор? Приведите примеры.
14.Таблица истинности простейшего сумматора и его схема.
15.Для чего используются карты Карно? Правила их построения.
16.Приведите пример применения карт Карно.
17.Постройте карты Карно для устройства, реализующего обнаружение первых шести простых чисел.
18.Постройте карты Карно для трехвходовой логической схемы, реализующей функцию F=3*x+2 (x – двоичный набор трех переменных).